Predicting Protein Interactions with Computational Geometry and Machine Learning Methods

Lukas Willy Bruhn
born 13th March 1994 in Kiel, Germany

Master Thesis Mathematics
Supervisor: Prof. Dr. Volkmar Liebscher
Second Supervisor: Pd. Dr. Christopher Horst Lillig

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

Institute of Mathematics and Computer Science
University of Greifswald

May 5, 2019

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Greifswald, May 5, 2019

Lukas Willy Bruhn

Ich arbeite immer zielgerichtet, manchmal kenne ich das Ziel nur nicht.

Contents

List of Figures

List of Tables

Chapter 1

Abstract

A common trend that can be observed in many scientific fields, is the rapid rate at which new data can be obtained. In order to organize and analyze these collections of data many new algorithms have been developed. The field of machine-learning makes great profit of the scale at which available data is growing.

3D-laser-scans are a popular tool for generating 3D-models automatically. A laser is used to measure the distance over a grid of the surface of an object. These distances are then used to generate a 3D-model of the surface. Measuring more distances lead to a higher resolution of the model. In figure 1.1 a point-cloud of the stanford-bunny is displayed. A typical task is to merge multiple scans into one model as shown in figure 1.2. This means translating and rotating the different point-clouds in order to establish a one-to-one correspondence between the points. This problem is called point-set-registration. For such rigid transformations algorithms such as the iterative-closest-point [12], [21] exist.

Non-rigid registrations form a more challenging problem. As Memoli points out in [18], *finding a notion of similarity that has some insensitivity to different poses of the same object* is important in such a case. One example of this nature is the articulation of an object. In figure 1.3 different poses of the same model are displayed [23]. Robust point matching (RPM) [11] is one approach to this problem.

It is widely believed that proteins with similar geometrical properties also have similar functional properties [16]. A more specific hypothesis, that will be examined in this thesis, is that proteins with similar *electrostatic fields* have similar functional properties [5]. In figure 1.4 the iso-surfaces of proteins of the glutaredoxin-family are displayed. In what follows we present an implementation that is capable of predicting protein-protein-interactions based on the electric iso-surface. We introduce this topic in more detail in section 2. In the proposed method we apply the ideas presented in [18], which are applicable to 3-D-point-clouds. What makes this implementation espe-

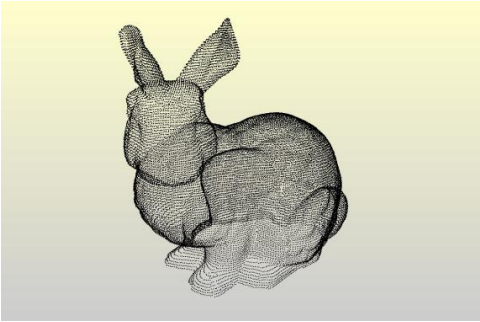


Figure 1.1: point-cloud-model of the stanford-bunny

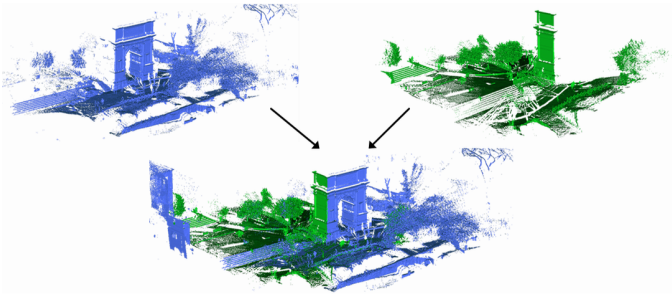


Figure 1.2: merging multiple scans

Figure 1.3: different articulations of the same model

cially attractive for biologists, is the fact that we generate our data from pure protein-data-bank-files (pdb), which can be downloaded freely from the pdb-databank [4]. The repository containing the implementation can be found here **https://github.com/WillyBruhn/PredictingProteinInteractions** and is freely available to the public running under the *GNU*-public-license.

This thesis is structured as follows. In section 2 the protein-specific problem is introduced in more detail. In section 3 the theory presented in [18] is reviewed. In section 4 one way of applying the methods from [18] to the proteins is described and problem-specific models are presented. In section 5 the theory of machine-learning as presented in [19] is reviewed and the protein-prediction-problem is stated in a formal way. In section 6 the models from

section 4 in combination with the machine-learner described in section 5 are tested.

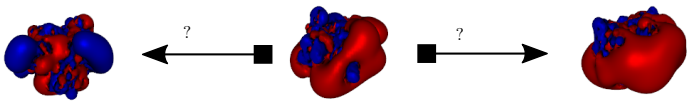


Figure 1.4: Isosurfaces of three different proteins

Chapter 2

Predicting Protein Interactions

2.1 Introduction

The idea of inferring functional similarity through geometrical similarity is not new. In [16] the authors propose that proteins with geometrical similar properties can have similar functional properties.

Thioredoxins form a family of proteins that contain about 100 amino-acids and a di-sulfid-bond as the active-center. Thioredoxins function as electron-donors [20] and have a high relevance in practically any known organism.

Lillig et. al. [5] proposed that the relevance of the geometrical complementarity of the surfaces of two interacting proteins is not only locally around the active site important but also that long range interactions around the surface play an important role. The analogy next to the key-lock-model and induced-fit-theory [17] is the following: In order for the key to work with a lock it also needs to be guided toward the lock. Protein iso-surfaces need not only to match well enough at the active site, but surface interactions far away from the active site are necessary to lead the active sites to each other.

In order to verify this hypothesis Lillig et. al. did the following: Given a set of proteins of which the capability to functionally replace each other in vitro, that means in a living cell,is known, the iso-surfaces are computed with VMD. Then Lillig claims that just with the naked eye the geometrical properties can be used to make an accurate prediction of the above mentioned functionality. Many experiments were conducted and one of special interest for this thesis, to which we will refer as the *106-redoxin-test-set*, was conducted in E.coli. Lillig achieved a relatively high accuracy with his method. However the method involves tideous steps and require an expert to manually examine each protein. When I joined Lilligs group, my first contribution

was to automatize the time-consuming steps, that are necessary to obtain an image of all the iso-surfaces of the proteins that one wants to examine (for more details see section 7.0.1). However this still needs an expert and hence the goal was to also automatize the prediction-step. A program capable of predicting functionality of proteins purely based on pdb-files could be used large-scale on databases, e.g. the protein-data-bank [4]. Another thinkable use-case could be to query a database for functional candidates of a given target-protein.

Also the proposed hypothesis of long-range-interactions can be put in mathematical models and can be examined.

2.2 Testset and Articulation of the Problem

In a previously conducted experiment by Lars .et .al in E.choli a gene-knock-down was performed. For E.choli the capability to produce Thrx is of vital importance, as it controls many regulatory reactions. Usually loosing the capability of Thrx-production leads to cell-death. In the conducted experiment the genes responsible for the production of Thrx were cut out of the genome and instead the genes for other proteins were inserted through gene-transfer. This was carried out for 106 different other proteins of the Gluta-redoxin-family. In the majority of cases the cells died, when they lost the capability to produce Thrx. In some cases however the cells continued to live. The conclusion was, that the replacing proteins were capable to replace Thrx in vital reactions. That means the proteins were *functionally similar*. Lars et. al. initially did not have an explanation as to why these proteins seemed to functionally replace Thrx. They conducted Lillig to take a look at this experiment. Lillig used his above described approach on this data-set blindfolded and retained a relatively high prediction-accuracy, just with the naked eye.

This biochemical experiment can be simplified as follows: We are given one molecule X and a set of molecules Y_i. Given one specific biochemical reaction that X is involved in, we want to predict which of the Y_i behave similarly to X in that biochemical reaction. For each Y_i a label f_i exists indicating if X and Y_i are similar. Since the labels are known, this problem is a supervised-learning-problem.

2.3 Goals of this Research

Motivated by the promising results from his above described approach Lillig asked both me and Felix Berens of the mathemtical institute of Greifswald to join his work-group for a master-thesis. The questions that Lillig was interested in are the following:

- Automatization of the *naked-eye-approach*

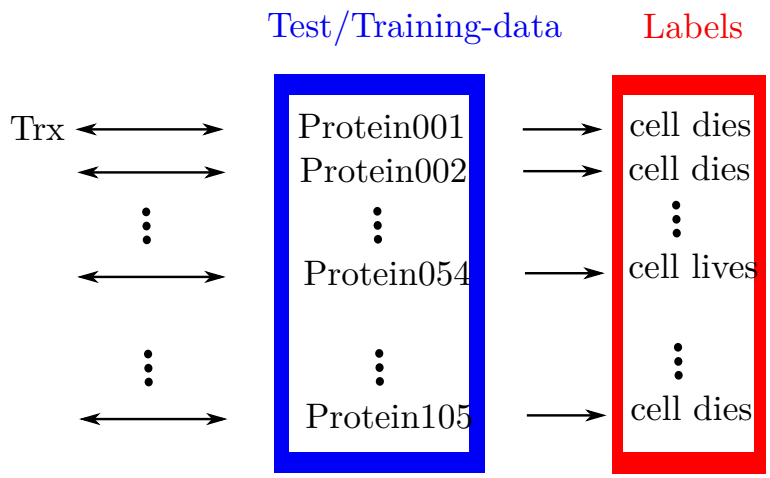


Figure 2.1: Schematic description of the test. Do the cells survive when Trx is replaced with some other Protein?

- Come up with a reasonable method for measuring the prediction-accuracy of a given method
- Establish a method that reaches an even higher prediction-accuracy
- Similarity allows for some local in-variance
- Role of the iso-surfaces
- Role of the active center
- Evaluation of the approach on additional data.

Chapter 3

Metric Geometry for Object Comparison

In this chapter the main definitions and concepts that are presented in [18] and form the basis for the approach are recalled. I recapitulate the theory in a condensed from as my colleague Felix Berens already examined the theory in his Master-thesis [3].

In summary the concepts relevant for this thesis from [18] are the following: The main concept is to compare the intrinsic distances of two objects rather than finding a one-to-one-correspondence of the points. In [18] the term *gromovization* is coined, which describes a deduction from the *Hausdorff-distance* to the *Grommov-Wasserstein-Distance* also called *earth-movers-distance* in informatics. Then a computationally fast lower bound (**FLB**) for the Grommov-Wasserstein-Distance is introduced. Again a computationally faster lower bound for the FLB is introduced. A procedure for sub-sampling points from the point-cloud is introduced. A 1-NN-classification-method is introduced. Lastly the method is tested on the publicly available data-set of 3D-models of different animals [23] to which will be referred as the *animal-test-set*.

I re-implemented the down-sampling-method and the classification-method in R. I implemented a *lower bound based on eccentricities* [18, Corollary 6.1 (Lower bound based on distribution of eccentricities)] (**DE**), which is a lower bound for the **FLB**, which itself is a lower bound for the *Grommov-Wasserstein-Distance*. It turns out that for the finite case ($|X|, |Y| < \infty$) it holds that **FLB** = **DE**. In [18] classification-performance was tested with the **FLB** on the *animal-test-set*. I tested the classification-performance of **DE** on the *animal-test-set*.

3.1 A Lower Bound for the Gromov-Wasserstein-distance

3.1 Definiton *[18]*
For a measurable map $f : X \mapsto Y$ between two compact metric spaces X and Y, and μ a measure on X, the push-forward-measure *$f_{\#\mu}$ on Y is given by*

$$f_{\#\mu}(A) := \mu(f^{-1}(A)) \quad for \quad A \in \mathcal{B}(Y) \tag{3.1}$$

where $\mathcal{B}(Y)$ is the Borel σ-algebra of Y.

3.2 Definiton *[18, Def. 5.1]*
A metric probability space *(mp-space for short, in [18] it is called metric measure-space) is a triple X, d_X, μ_X where*

- *(X, d_X) is a compact metric space.*
- *μ_X is a Borel probability measure on X i.e., $\mu_X(X) = 1$, and μ_X has full support:* $\text{supp}[\mu_X] = X$.

The triple (X, d_X, μ_X) will be denoted by $\mathbb{X}$. The reason for imposing $\mu_X(X) = 1$ is that one thinks of μ_X as a modelization of the acquisition process or sampling procedure of an object. Two mp-spaces (X, d_X, μ_X) and (Y, d_Y, μ_Y) are called *isomorphic* iff there exists an isometry $\Psi : X \to Y$ such that $\Psi_{\#}\mu_X = \mu_Y$. Furthermore, by $\mathcal{G}_w$ the collection of all mp-spaces will be denoted.

3.3 Definiton *(Measure coupling) [18, Def. 5.6]*
Given two metric measure spaces (X, d_X, μ_X) and (Y, d_Y, μ_Y) one says that a measure μ on the product space $X \times Y$ is a coupling *of μ_X and μ_Y iff*

$$\mu(A \times Y) = \mu_X(A), \quad and \quad \mu(X \times A') = \mu(A') \tag{3.2}$$

for all measurable sets $A \subset X, A' \subset Y$. Denote by $\mathcal{M}(\mu_X, \mu_Y)$ the set of all couplings of μ_X and μ_Y.

3.4 Definiton *[18]*
For metric spaces (X, d_X) and (Y, d_Y) let $\Gamma_{X,Y} : (X \times Y) \times (X \times Y) \mapsto \mathbb{R}^+$ be given by

$$\Gamma_{X,Y}(x, y, x'y') := |d_X(x, x') - d_Y(y, y')|. \tag{3.3}$$

3.5 Definiton *[18, Def. 5.7]*
For $\infty \geq p \geq 1$ one defines the Gromov-Wassertein-distance *$\mathfrak{D}_p$ between two mp-spaces $\mathbb{X}$ and $\mathbb{Y}$ by*

$$\mathfrak{D}_p(\mathbb{X}, \mathbb{Y}) := \inf_{\mu \in \mathcal{M}(\mu_X, \mu_Y)} \mathbf{J}_p(\mu) \tag{3.4}$$

where for $p \in [1, \infty)$ *and* $\mu \in \mathcal{M}(\mu_X, \mu_Y)$

$$\mathbf{J}_p(\mu) := \frac{1}{2}\Big(\int_{X\times Y}\int_{X\times Y}(\Gamma_{X,Y}(x,y,x',y'))^p \mu(dx\times dy)\mu(dx'\times dy')\Big)^{\frac{1}{p}}$$
$$\Big(= \frac{1}{2}||\Gamma_{X,Y}||_{L^P(\mu\otimes\mu_Y)}\Big) \tag{3.5}$$

and $p = \infty$

$$\mathbf{J}_\infty(\mu) := \frac{1}{2}\sup_{\substack{x,x'\in X\\ y,y'\in Y\\ s.t.(x,y),(x',y')\in R(\mu)}} \Gamma_{X,Y}(x,y,x',y')\Big(= \frac{1}{2}||\Gamma_{X,Y}||^\infty_L(R(\mu)\times R(\mu)).\tag{3.6}$$

3.6 Lemma *[18, Lemma. 6.1]*
Let (X, d_X, μ_X) *and* (Y, d_Y, μ_Y) *be two mp-spaces in* $\mathcal{G}_\omega$. *Let* $f : X \mapsto \mathbb{R}$ *and* $g : Y \mapsto \mathbb{R}$ *be continuous and* $\phi : \mathbb{R} \mapsto (0, \infty)$ *be convex. Then*

$$\inf_{\mu\in\mathcal{M}}(\mu_X,\mu_Y)\int_{X\times Y}\phi(f(x)-g(y))\mu(dx\times dy) \geq \int_0^1 \phi(F^{-1}(t)-G^{-1}(t))dt \tag{3.7}$$

where $F(t) := \mu_X\{x \in X | f(x) \leq t\}$ *and* $G(t) := \mu_Y\{y \in Y | f(y) \leq t\}$ *are the distributions of* f *and* g, *respectively, and their generalized inverses under* μ_X *and* μ_Y, *respectively, are defined as:*

$$F^{-1}(t) = \inf\{u \in \mathbb{R} | F(u) > t\}.$$

Furthermore, when $\phi(u) = |u|, u \in \mathbb{R}$, *one can dispense with the inverses:*

$$\inf_{\mu\in\mathcal{M}}(\mu_X,\mu_Y)\int_{X\times Y}|\phi(f(x)-g(y))|\mu(dx\times dy) \geq \int_{\mathbb{R}}\phi(F(u)-G(u))du \tag{3.8}$$

Proof: Fix $\mu \in \mathcal{M}(\mu_X, \mu_Y)$. Let $h : X \times Y \Rightarrow \mathbb{R}^2$ given by $h = (f, g)$ and consider the measure $\nu = h_{\#}\mu$ on $\mathbb{R}^2$. By Theorem 4.1.11 of [](applied to $T : X \times Y \Rightarrow \mathbb{R} \times \mathbb{R}, (x, y) \mapsto (f(x), g(y)))$ one has

$$\int_{X\times Y}\phi(f(x)-g(y))\mu(d_X\times d_Y) = \int_{\mathbb{R}\times\mathbb{R}}\phi(t-s)\nu(dt\times ds). \tag{3.9}$$

Now, $\nu(I \times \mathbb{R}) = \mu(f^{-1}(I) \times g^{-1}(\mathbb{R}) = \mu(f^{-1}(I) \times Y) = \mu_X(f^{-1}(I))$, for any $I \in \mathcal{B}(\mathbb{R})$. Similarly, $\nu(\mathbb{R} \times J) = \mu_Y(g^{-1}(J))$ for any $J \in \mathcal{B}(\mathbb{R})$. Hence from the equality above,

$$\int_{X\times Y}\phi(f(x)-g(y))\mu(dx\times dy) \geq \inf_{\nu\in\mathcal{M}(f_{\#}\mu_X, g_{\#}\mu_Y)}\int_{\mathbb{R}\times\mathbb{R}}\phi(t-s)\nu(dt\times ds). \tag{3.10}$$

The conclusion follows from results on the transportation problem on the real line, see Remark 2.19 in [24], and then from the fact that $\mu \in \mathcal{M}(\mu_X, \mu_Y)$ was arbitrary and the right-hand side does not depend on it.

□

3.7 Definiton *[18, Def. 5.3]*
Given $p \in [1, \infty]$ and an mp-space (X, d_X, μ_X) define the p-eccentricity function *of X by*

$$s_{X,p} : X \to \mathbb{R}^+ \quad \textit{given by} \quad x \mapsto \Big(\int_X d_X(x, x')^p \mu_X(dx')\Big)^{\frac{1}{p}} (= ||d_X(x, \cdot)||_{L^p(\mu_X)}) \tag{3.11}$$

for $1 \leq p \leq \infty$, and by

$$s_{X,\infty} : X \to \mathbb{R}^+ \quad \textit{given by} \quad x \mapsto \sup_{x' \in supp[\mu_X]} d_X(x, x') \tag{3.12}$$

for $p = \infty$.

Let $S_{X,p} : \mathbb{R} \mapsto [0, 1]$ be given by $t \mapsto \mu_X(\{x \in X | s_{X,p}(x) \leq t\})$, i.e., $S_{X,p}$ is the distribution function of $s_{X,p}$ under μ_X.

3.8 Definiton *(Distribution of Eccentricities)*
For $\mathbb{X}, \mathbb{Y} \in \mathcal{G}_\omega$ define:

$$\boldsymbol{DE}(\mathbb{X}, \mathbb{Y}) := \frac{1}{2}\Big(\int_{\mathbb{R}} |S_{X,1}(t) - S_{Y,1}(t)| dt\Big). \tag{3.13}$$

3.9 Definiton *(First Lower Bound) [18, Def. 6.1]*
For $\mathbb{X}, \mathbb{Y} \in \mathcal{G}_\omega$ define for $p \in [1, \infty)$:

$$\boldsymbol{FLB}_p(\mathbb{X}, \mathbb{Y}) := \frac{1}{2} \inf_{\mu \in \mathcal{M}(\mu_X, \mu_Y)} \Big(\int_{X \times Y} |s_{X,p}(x) - s_{Y,p}(y)|^p \mu(dx \times dy)\Big)^{\frac{1}{p}}. \tag{3.14}$$

$\boldsymbol{FLB}_1$ will be called $\boldsymbol{FLB}$.

3.10 Proposition *[18, Prop. 6.1]*
Let $X, Y \in \mathcal{G}_\omega$ and $p \in [1, \infty]$. Then,

$$\mathfrak{D}_p(X, Y) \geq \boldsymbol{FLB}_p(X, Y). \tag{3.15}$$

□

Note that for $1 \leq p < \infty$, solving for $\mathbf{FLB}_p$ leads to a *Mass Transportation Problem* for the cost $c(x, y) := |s_{X,p}(x) - s_{Y,p}(y)|^p$.

Then, invoking Lemmma 3.6 with $\phi(u) = |u|^p$ one obtains

3.11 Corollary *(Lower bound based on distribution of eccentricities) [18, Cor. 6.1]*
For $X, Y \in \mathcal{G}_\omega$,

$$\boldsymbol{FLB}(X,Y) \geq \boldsymbol{DE}. \tag{3.16}$$

3.12 Remark *(Matching measures between finite spaces) [18, Rem. 2.2]*
When $X, Y \in \mathcal{C}_\omega(Z)$ are finite, say $n_X = |X|$ and $n_Y = |Y|$, then $\mathcal{M}(\mu_X, \mu_Y)$ is composed of matrices with non-negative entries of size $n_X \times n_Y$ satisfying $n_X + n_Y$ linear constraints:

$$\sum_{x \in X} \mu(x,y) = \mu_Y(y) \quad \forall y \in Y \quad \text{and} \quad \sum_{y \in Y} \mu(x,y) = \mu_X(x) \quad \forall x \in X. \tag{3.17}$$

3.2 My Theoretical Contribution

3.2.1 DE = FLB

In this section I proof that for the finite case $|X|, |Y| < \infty$ and $\phi := |\cdot|$ it holds **DE** = **FLB**.

3.13 Lemma
In the situation of Lemma 3.6 let $|X| = n$ and $|Y| = m$. With $\phi := |\cdot|$ it holds:

$$\inf_{\mu \in \mathcal{M}} (\mu_X, \mu_Y) \int_{X \times Y} |f(x) - g(y)| \mu(dx \times dy) = \int_{\mathbb{R}} |F(u) - G(u)| du. \tag{3.18}$$

Proof: see below.

3.14 Corollary *(DE = FLB)*
Let $|X| = n$ and $|Y| = m$. Then it holds:

$$\boldsymbol{FLB}(\mathbb{X}, \mathbb{Y}) = \boldsymbol{DE}(\mathbb{X}, \mathbb{Y}). \tag{3.19}$$

Proof: Analogously to Corollary 3.11. □

DE = FLB with arbitrary measure, direct proof

In what follows some notations are introduced that lead up to prove the equality of the **FLB** and **DE** in the finite case. In the situation of Lemma

3.6 and the case that $\phi := |\cdot|$ let X and Y be permuted such that $f(x_1) \leq f(x_2) \leq \cdots$ and $g(y_1) \leq g(y_2) \leq, \cdots$. Then denote with

$$F_i := \sum_{k=1}^{i} \mu_X(x_k) \tag{3.20}$$

$$G_j := \sum_{k=1}^{j} \mu_Y(y_k). \tag{3.21}$$

That means $F_i = F(t)$ for $t \in [f(x_i), f(x_{i+1}))$ and $G_j = G(t)$ for $t \in [g(y_j), g(y_{j+1}))$.

Now define

$$H : \mathbb{R}^2 \mapsto \mathbb{R} \tag{3.22}$$

$$H(s,t) := \min\{F(s), G(t)\} \tag{3.23}$$

$$H_{ij} := H(s,t) \quad \text{for } s \in [f(x_i), f(x_{i+1})), \quad t \in [g(y_j), g(y_{j+1})) \tag{3.24}$$

$$\mu_{ij} := H_{ij} - \max\{H_{i-1j}, H_{ij-1}\} \tag{3.25}$$

$$\mu : X \times Y \mapsto [0,1] \tag{3.26}$$

$$\mu(x_i, y_j) := \mu_{i,j}. \tag{3.27}$$

3.15 Lemma
Let $\mu_X(x) > 0 \quad \forall x$ and $\mu_Y(y) > 0 \quad \forall y$. Then

$$H_{i,j+1}, < H_{i+1,j} \Rightarrow \mu_{i',j+1} = 0 \quad \forall i' \leq i \tag{3.28}$$

$$H_{i,j+1}, > H_{i+1,j} \Rightarrow \mu_{i+1,j'} = 0 \quad \forall j' \leq j \tag{3.29}$$

$$H_{i,j+1}, = H_{i+1,j} \Rightarrow \mu_{i',j'} = 0 \quad \forall j' \leq j, \quad \forall i' \leq i. \tag{3.30}$$

Proof: *It is sufficient to prove 3.28 since 3.29 follows by switching the roles of F and G. Since $\mu_X(x_i) > 0$ it follows that $F_i < F_{i+1}$ and since $\mu_Y(y_j) > 0$ it follows that $G_j < G_{j+1}$.*

Assume $F_{i'} \leq G_j \quad \forall i' \leq i$. Then

$$\mu_{i',j+1} = H_{i',j+1} - \max\{H_{i',j}, H_{i'-1,j+1}\} \tag{3.31}$$

$$= \min\{F_{i'}, G_{j+1}\} - \max\{\min\{F_{i'}, G_j\}, \min\{F_{i'-1}, G_{j+1}\}\} \tag{3.32}$$

$$= F_{i'} - \max\{F_{i'}, F_{i'-1}\} \tag{3.33}$$

$$= F_{i'} - F_{i'} = 0. \tag{3.34}$$

Assume $H_{i,j+1} < H_{i+1,j}$. From a case distinction one obtains that:

$$H_{i,j+1} < H_{i+1,j} \Rightarrow \min\{F_i, F_{i+1}, G_j, G_{j+1}\} = F_i \tag{3.35}$$

since

$$1.) F_i < G_j \tag{3.36}$$
$$\Rightarrow H_{i+1,j} = \min\{F_{i+1}, G_j\} > F_i = \min\{F_i, G_{j+1}\} = H_{i,j+1} \quad \surd \tag{3.37}$$
$$2.) F_i > G_j \tag{3.38}$$
$$\Rightarrow H_{i+1,j} = \min\{F_{i+1}, G_j\} = G_j < \min\{F_i, G_{j+1}\} = H_{i,j+1} \quad \lightning \tag{3.39}$$
$$3.) F_i = G_j \tag{3.40}$$
$$\Rightarrow H_{i+1,j} = \min\{F_{i+1}, G_j\} = \min\{F_{i+1}, F_i\} = F_i \tag{3.41}$$
$$= G_j = \min\{G_j, G_{j+1}\} = \min\{F_i, G_{j+1}\} = H_{i,j+1} \quad \lightning \tag{3.42}$$
$$\tag{3.43}$$

Therefore $F_{i'} < G_j \quad \forall i' \leq i$ *in the case that* $H_{i,j+1} < H_{i+1,j}$. *Analogously for* $H_{i,j+1} > H_{i+1,j}$. *for For the case that* $H_{i,j+1}, = H_{i+1,j}$ *it follows* $F_i = G_j$ *(see case distinction above). Therefore in all three cases* $H_{i,j+1} < H_{i+1,j}$, $H_{i,j+1} > H_{i+1,j}$ *and* $H_{i,j+1} = H_{i+1,j}$ *it holds* $F_{i'} \leq G_j \quad \forall i' \leq i$. *With 3.31 it follows* $\mu_{i',j+1} = 0 \quad \forall i' \leq i$ *respectively* $\mu_{i+1,j'} = 0 \quad \forall j' \leq j$.

□

3.16 Lemma
Let $n' \leq n$ *and* $m' \leq m$:

$$\sum_{i=1}^{n'} \sum_{j=1}^{m'} \mu_{ij} = H_{n'm'}. \tag{3.44}$$

Proof: $H_{1,j} = \min\{F_1, G_j\}$. *Proof by Induction: Anchor:*

$$H_{1,1} = \min\{F_1, G_1\}, \quad \mu_{1,1} = H_{1,1} - \max\{0, 0\} = H_{1,1}. \tag{3.45}$$

Step:

$$\sum_{k=1}^{j} \mu_{1,k} = H_{1,j} \Rightarrow \sum_{k=1}^{j+1} = H_{1,j+1}. \tag{3.46}$$

$$\mu_{1,j+1} = H_{1,j+1} - \max\{H_{1,j}, 0\} \tag{3.47}$$
$$= H_{1,j+1} - H_{1,j} \tag{3.48}$$
$$= H_{1,j+1} - \sum_{k=1}^{j} \mu_{1,k} \tag{3.49}$$
$$\Leftrightarrow \tag{3.50}$$
$$\mu_{1,j+1} + \sum_{k=1}^{j} \mu_{1,k} = H_{1,j+1} \tag{3.51}$$
$$\sum_{k=1}^{j+1} \mu_{1,k} = H_{1,j+1} \tag{3.52}$$

Step end.

It follows:

$$\sum_{k=1}^{j} \mu_{1,k} = H_{1,j} \quad \forall j \tag{3.53}$$

$$\sum_{r=1}^{i} \mu_{r,1} = H_{i,1} \quad \forall i. \tag{3.54}$$

Induction step 2:

$$\sum_{r=1}^{i+1}\sum_{k=1}^{j} \mu_{r,k} = H_{i+1,j} \Rightarrow \sum_{r=1}^{i+1}\sum_{k=1}^{j+1} \mu_{r,k} = H_{i+1,j+1}. \tag{3.55}$$

Proof of the induction 2:

$$\mu_{i+1,j+1} = H_{i+1,j+1} - \max\{H_{i,j+1}, H_{i+1,j}\} \tag{3.56}$$

Assume $H_{i,j+1} \leq H_{i+1,j}$, then:

$$\mu_{i+1,j+1} = H_{i+1,j+1} - \sum_{r=1}^{i+1}\sum_{k=1}^{j} \mu_{r,k} \tag{3.57}$$

$$\mu_{i+1,j+1} + \sum_{r=1}^{i+1}\sum_{k=1}^{j} \mu_{r,k} = H_{i+1,j+1} \tag{3.58}$$

$$\sum_{r=1}^{i+1}\sum_{k=1}^{j+1} \mu_{r,k} - \sum_{r=1}^{i} \mu_{r,j+1} = H_{i+1,j+1}. \tag{3.59}$$

With Lemma 3.15 it follows $\sum_{r=1}^{i} \mu_{r,j+1} = 0$.
Therefore $\sum_{r=1}^{i+1}\sum_{k=1}^{j+1} \mu_{r,k} = H_{i+1,j+1}$. Analogously for $Hi, j+1 > H_{i+1,j}$
□

3.17 Lemma
$\mu \in \mathcal{M}(\mu_X, \mu_Y)$

Proof: *Let $A = \{x_i\}$. It has to be shown that:*

$$\mu(A \times Y) = \mu_X(A) \tag{3.60}$$

$$\Leftrightarrow \sum_{j=1}^{m} \mu_{ij} = \mu_X(A) = \mu_X(\{x_i\}) = F_i - F_{i-1}. \tag{3.61}$$

With Lemma 3.16 it follows:

$$Q := \sum_{k=1}^{i}\sum_{j=1}^{m} \mu_{k,j} = H_{i,m} = \min\{F_i, G_m\} \tag{3.62}$$

$$= \min\{F_i, \max_j\{G_j\}\} \tag{3.63}$$

$$= \min\{F_i, \max_r\{F_r\}\} = F_i. \tag{3.64}$$

$$P := \sum_{k=1}^{i-1}\sum_{j=1}^{m} \mu_{k,j} = H_{i-1,m} = \min\{F_i i - 1, G_m\} = F_i i - 1. \tag{3.65}$$

Then:

$$F_i - F_{i-1} = Q - P = \sum_{k=1}^{i}\sum_{j=1}^{m} \mu_{k,j} - \sum_{k=1}^{i-1}\sum_{j=1}^{m} \mu_{k,j} = \sum_{k=i}^{i}\sum_{j=1}^{m} \mu_{k,j} = \sum_{j=1}^{m} \mu_{i,j}. \tag{3.66}$$

Any set $A' \subseteq X$ is a finite union of sets like A ($A' = \bigcup_{i\in I} A_i$). For the sets in Y analogously.
Therefore $\mu \in \mathcal{M}(\mu_X, \mu_Y)$.

□

Proof of 3.13 ((DE = FLB)): Denote with $d_{i,j}(t) := \mathbb{1}_{[f(x_i),\infty)}(t) - \mathbb{1}_{[g(y_j),\infty)}(t)$. Let $t \in \mathbb{R}$. Then $\forall i, j, k, l$ with $\mu_{i,j} > 0$ and $\mu_{k,l} > 0$ it holds that $\operatorname{sign}(d_{i,j}(t)) = \operatorname{sign}(d_{k,l}(t))$.
Proof by contradiction: Assume $\operatorname{sign}(d_{i,j}(t)) \neq \operatorname{sign}(d_{k,l}(t))$ or more specifically assume $d_{i,j}(t) > 0$ and $d_{k,l}(t) < 0$. Then it follows $f(x_i) \leq t < g(y_j)$ and $g(y_l) \leq t < f(x_k)$. It follows that $i < k, l < j$. By definition of μ it holds

$$\mu_{k,j} = H_{k,j} - \max\{H_{k-1,j}, H_{k,j-1}\}. \tag{3.67}$$

Assume $H_{k-1,j} \geq H_{k,j-1}$. Then with Lemma 3.15 it follows $\mu_{k,l} = 0$, which is a contradiction.
Assume $H_{k-1,j} \leq H_{k,j-1}$. Then with Lemma 3.15 it follows $\mu_{i,j} = 0$, which is also a contradiction. Therefore it holds that $\forall i, j, k, l$ with $\mu_{i,j} > 0$ and

$\mu_{k,l} > 0$, $\text{sign}(d_{i,j}(t)) = \text{sign}(d_{k,l}(t))$. With this it follows:

$$\int_{\mathbb{R}} |F(t) - G(t)|dt = \int_{\mathbb{R}} |\sum_{i=1}^{n} \mu_X(x_i) \cdot \mathbb{1}_{[f(x_i),\infty)}(t) - \sum_{j=1}^{m} \mu_Y(y_j) \cdot \mathbb{1}_{[g(y_j),\infty)}(t)|dt \tag{3.68}$$

$$= \int_{\mathbb{R}} |\sum_{i=1}^{n}\sum_{j=1}^{m} \mu_{i,j} \cdot \mathbb{1}_{[f(x_i),\infty)}(t) - \sum_{j=1}^{m}\sum_{i=1}^{n} \mu_{i,j} \cdot \mathbb{1}_{[g(y_j),\infty)}(t)|dt \tag{3.69}$$

$$\int_{\mathbb{R}} |\sum_{i=1}^{n}\sum_{j=1}^{m} \mu_{i,j} \cdot (\mathbb{1}_{[f(x_i),\infty)}(t) - \mathbb{1}_{[g(y_j),\infty)}(t))|dt \tag{3.70}$$

$$= \int_{\mathbb{R}} \sum_{i=1}^{n}\sum_{j=1}^{m} \mu_{i,j} \cdot |\mathbb{1}_{[f(x_i),\infty)}(t) - \mathbb{1}_{[g(y_j),\infty)}(t)|dt \tag{3.71}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} \mu_{i,j} \cdot \int_{\mathbb{R}} |\mathbb{1}_{[\min\{f(x_i),g(y_j)\}, \max\{f(x_i),g(y_j)\})}(t)|f(x_i) - g(y_j)| \tag{3.72}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} \mu_{i,j} \cdot |f(x_i) - g(y_j)| \tag{3.73}$$

$$= \int_{X\times Y} |f(x) - g(y)|\mu(d_X \times d_Y). \tag{3.74}$$

Lemma 6.1. showed that:

$$\inf_{\mu' \in \mathcal{M}}(\mu_X, \mu_Y) \int_{X\times Y} |f(x) - g(y)|\mu'(dx \times dy) \geq \int_{\mathbb{R}} |F(t) - G(t)|dt. \tag{3.75}$$

Since $\mu \in \mathcal{M}$ (Lemma 3.17), and

$$\int_{\mathbb{R}} |F(t) - G(t)|dt = \int_{\mathbb{X}\times\mathbb{Y}} |f(x) - g(y)|\mu(d_X \times d_Y) \tag{3.76}$$

it follows that

$$\inf_{\mu' \in \mathcal{M}}(\mu_X, \mu_Y) \int_{X\times Y} |f(x) - g(y)|\mu'(dx \times dy) = \int_{X\times Y} |f(x) - g(y)|\mu(d_X \times d_Y). \tag{3.77}$$

Therefore

$$\inf_{\mu' \in \mathcal{M}}(\mu_X, \mu_Y) \int_{X\times Y} |f(x) - g(y)|\mu'(dx \times dy) = \int_{\mathbb{R}} |F(t) - G(t)|dt. \tag{3.78}$$

□

DE = FLB with arbitrary measure, proof with literature

3.18 Theorem *(Optimal transportation theorem for a quadratic cost on $\mathbb{R}$) [24, Section 2.2 Theorem 2.18]*
Let μ_F, μ_G be two probability measures on $\mathbb{R}$, with respective cumulative distribution functions F and G. Let Π be the probability measure on $\mathbb{R}^2$ with joint two-dimensional cumulative distribution function

$$H(s,t) = \min\{F(s), G(t)\}. \tag{3.79}$$

Then, Π belongs to $\mathcal{M}(\mu_F, \mu_G)$, and is optimal in the Kantorovich transportation problem between μ_F and μ_G for the quadratic cost function $c(r,s) = |r-s|^2$. Moreover, the value of the optimal transportation cost is

$$\mathcal{T}_2(\mu_F, \mu_G) = \int_0^1 |F^{-1}(t) - G^{-1}(t)|^2 dt. \tag{3.80}$$

3.19 Remark *[24, Section 2.2 Remark 2.19]*

*(i) The **Hoeffding-Frechet theorem** states that a non-negative function H on $\mathbb{R}^2$, non-decreasing and right-continuous in each argument, defines a probability measure π on $\mathbb{R}^2$ with given marginals μ, ν if and only if*

$$\forall (x,y) \in \mathbb{R}^2, \quad F(x) + G(y) - 1 \leq H(x,y) \leq \min(F(x), G(y)),$$

where F and G are the cumulative distribution functions associated with μ and ν respectively.

(ii) The π constructed in 3.18 is optimal whatever the convex cost. More precisely, π is optimal as soon as the cost function $c(x,y)$ takes the form $c(x-y)$, where c is a convex nonevative symmetric function on $\mathbb{R}$. In this case the optimal transportation cost is

$$\mathcal{T}_1(\mu_F, \mu_G) = \int_0^1 |F^{-1}(t) - G^{-1}(t)| dt. \tag{3.81}$$

(iii) The total transportation cost associated with the cost function $c(x,y) = |x-y|$ is

$$\mathcal{T}_1(\mu_u, \mu_v) = \int_0^1 |U^{-1}(t) - V^{-1}(t)| dt = \int_{\mathbb{R}} |U(t) - V(t)| dt. \tag{3.82}$$

Proof of 3.13 (DE = FLB):
Define $\mu_F := f_{\#\mu_X}$ and $\mu_G := g_{\#\mu_Y}$. Then

$$F(s) = \mu_F((-\infty, s]) = f_{\#\mu_X}((-\infty, s]) = \mu_X(f^{-1}((-\infty, s]))$$

and

$$G(t) = \mu_G((-\infty, t]) = g_{\#\mu_Y}((-\infty, t]) = \mu_Y(g^{-1}((-\infty, t])).$$

Then it follows:

$$\begin{aligned}\int_{R(s_0,t_0)} dh_{\#\mu} =& h_{\#\mu}(R(s_0,t_0))\\ =& \mu(h^{-1}(R(s_0,t_0)))\\ =& \mu(h^{-1}(\{(s,t) : s \leq s_0, t \leq t_0\}))\\ =& \mu(\{(x_i,y_j) : f(x_i) \leq s_0, g(y_j) \leq t_0\}).\end{aligned}$$

Then for $n' := \max\{i : f(x_i) \leq s_0\}$ and $m' := \max\{j : g(y_j) \leq t_0\}$ it follows:

$$\mu(\{(x_i,y_j) : f(x_i) \leq s_0, g(y_j) \leq t_0\}) = \sum_{n'}^{i=1}\sum_{m'}^{j=1} \mu_{i,j}.$$

With Lemma 3.16 it follows that $\sum_{n'}^{i=1}\sum_{m'}^{j=1} \mu_{i,j} = H_{n',m'}$. Therefore

$$\sum_{n'}^{i=1}\sum_{m'}^{j=1} \mu_{i,j} = H_{n',m'} = H(s_0,t_0).$$

Therefore H is the cumulative distribution function of $h_{\#\mu}$. Therefore with Theorem 3.18 and $\pi := h_{\#\mu}$ it follows that $\mu \in \mathcal{M}(\mu_X,\mu_Y)$ and is optimal in the Kantorovich transportation problem, that means

$$\inf_{\mu' \in \mathcal{M}}(\mu_X,\mu_Y)\int_{X\times Y} |f(x)-g(y)|^2 \mu'(dx\times dy) = \int_{X\times Y} |f(x)-g(y)|^2 \mu(dx\times dy)$$

and

$$\int_{X\times Y} |f(x)-g(y)|^2 \mu(dx\times dy) = \int_0^1 |F^{-1}(t)-G^{-1}(t)|^2 dt.$$

.

With Remark 3.19(ii) it follows

$$\inf_{\mu' \in \mathcal{M}}(\mu_X,\mu_Y)\int_{X\times Y} |f(x)-g(y)|\mu'(dx\times dy) = \int_{X\times Y} |f(x)-g(y)|\mu(dx\times dy).$$

and

$$\int_{X\times Y} |f(x)-g(y)|\mu(dx\times dy) = \int_0^1 |F^{-1}(t)-G^{-1}(t)| dt.$$

.

Furthermore with Remark 3.19(*iii*) it follows

$$\int_{X\times Y} |f(x) - g(y)|\mu(dx \times dy) = \int_{\mathbb{R}} |F(t) - G(t)|dt.$$

Therefore it follows

$$\inf_{\mu' \in \mathcal{M}(\mu_X, \mu_Y)} \int_{X\times Y} |f(x) - g(y)|\mu'(dx \times dy) = \int_{\mathbb{R}} |F(t) - G(t)|dt.$$

□

3.2.2 Some Examples

3.20 Example 2 *(A Transportation map and $F - G$)*

$$\mu_X = \frac{1}{6} \cdot \begin{pmatrix} 1 & 4 & 1 \end{pmatrix} \tag{3.83}$$

$$f = \begin{pmatrix} 10 & 10 & 100 \end{pmatrix}$$

$$F = \frac{1}{6} \cdot \begin{pmatrix} 1 & 5 & 6 \end{pmatrix} \tag{3.84}$$

$$\mu_Y = \frac{1}{6} \cdot \begin{pmatrix} 1 & 3 & 2 \end{pmatrix} \tag{3.85}$$

$$g = \begin{pmatrix} 1 & 5 & 7 \end{pmatrix}$$

$$G = \frac{1}{6} \cdot \begin{pmatrix} 1 & 4 & 6 \end{pmatrix} \tag{3.86}$$

$$H = \frac{1}{6} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 4 & 5 \\ 1 & 4 & 6 \end{pmatrix} \tag{3.87}$$

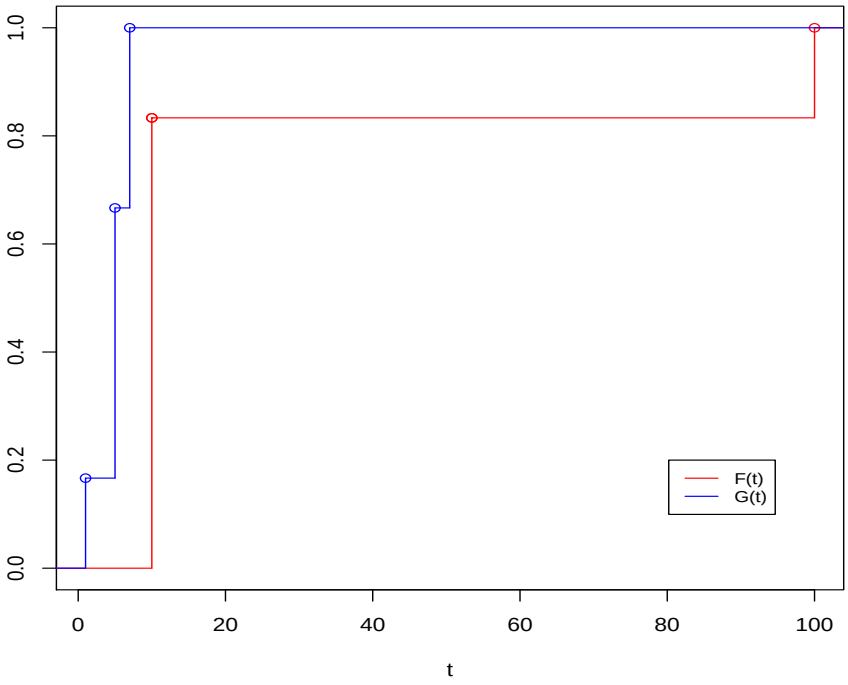


Figure 3.1: Distributions of F and G

$$\mu = \frac{1}{6} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.88}$$

The left-hand-side of the equation looks like this:

$$
\begin{aligned}
&\int_{X\times Y} |f(x) - g(y)|\mu(d_x \times d_y) \\
&= \sum_{i,j} \mu_{i,j} \cdot |f_i - g_j| \\
&= \mu_{1,1} \cdot |f_1 - g_1| + \mu_{2,2} \cdot |f_2 - g_2| + \mu_{2,3} \cdot |f_2 - g_3| + \mu_{3,3} \cdot |f_3 - g_3| \\
&= \frac{1}{6} \cdot |10 - 1| + \frac{3}{6} \cdot |10 - 5| + \frac{1}{6} \cdot |10 - 7| + \frac{1}{6} \cdot |100 - 7| \\
&= \frac{1}{6} \cdot |9| + \frac{3}{6} \cdot |5| + \frac{1}{6} \cdot |3| + \frac{1}{6} \cdot |93| \\
&= \frac{1}{6} \cdot (9 + 15 + 3 + 93) \\
&= \frac{1}{6} \cdot 120 \\
&= 20
\end{aligned}
$$

.

The right-hand-side of the equation looks like this:

$$
\begin{aligned}
&\int_t |F(t) - G(t)| \\
&= \sum_{i=2}^{n_X+n_Y} |t_i - t_{i-1}| \cdot |F(t_{i-1}) - G(t_{i-1})| \\
&= |t_2 - t_1| \cdot |F(t_1) - G(t_1)| + |t_3 - t_2| \cdot |F(t_2) - G(t_2)| \\
&+ |t_4 - t_3| \cdot |F(t_3) - G(t_3)| + |t_5 - t_4| \cdot |F(t_4) - G(t_4)| \\
&= |5 - 1| \cdot |F(1) - G(1)| + |7 - 5| \cdot |F(5) - G(5)| \\
&+ |10 - 7| \cdot |F(7) - G(7)| + |100 - 10| \cdot |F(10) - G(10)| \\
&= |5 - 1| \cdot \frac{1}{6} \cdot |0 - 1| + |7 - 5| \cdot \frac{1}{6} \cdot |0 - 4| \\
&+ |10 - 7| \cdot \frac{1}{6} \cdot |0 - 6| + |100 - 10| \cdot \frac{1}{6} \cdot |5 - 6| \\
&= |4| \cdot \frac{1}{6} \cdot |-1| + |2| \cdot \frac{1}{6} \cdot |-4| \\
&+ |3| \cdot \frac{1}{6} \cdot |-6| + |90| \cdot \frac{1}{6} \cdot |-1| \\
&= 0.666666666666667 + 1.33333333333333 \\
&+ 3 + 15 \\
&= 20
\end{aligned}
$$

.

3.21 Example 3 *(A Transportation map and $F - G$)*

$$\mu_X = \frac{1}{20} \cdot \begin{pmatrix} 2 & 1 & 1 & 4 & 3 & 8 & 1 \end{pmatrix} \tag{3.89}$$

$$f = \begin{pmatrix} 5 & 10 & 11 & 14 & 19 & 25 & 100 \end{pmatrix}$$

$$F = \frac{1}{20} \cdot \begin{pmatrix} 2 & 3 & 4 & 8 & 11 & 19 & 20 \end{pmatrix} \tag{3.90}$$

$$\mu_Y = \frac{1}{20} \cdot \begin{pmatrix} 4 & 1 & 1 & 6 & 8 \end{pmatrix} \tag{3.91}$$

$$g = \begin{pmatrix} 1 & 5 & 7 & 14 & 35 \end{pmatrix}$$

$$G = \frac{1}{20} \cdot \begin{pmatrix} 4 & 5 & 6 & 12 & 20 \end{pmatrix} \tag{3.92}$$

$$H = \frac{1}{20} \cdot \begin{pmatrix} 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 4 & 5 & 6 & 8 & 8 \\ 4 & 5 & 6 & 11 & 11 \\ 4 & 5 & 6 & 12 & 19 \\ 4 & 5 & 6 & 12 & 20 \end{pmatrix} \tag{3.93}$$

$$\mu = \frac{1}{20} \cdot \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.94}$$

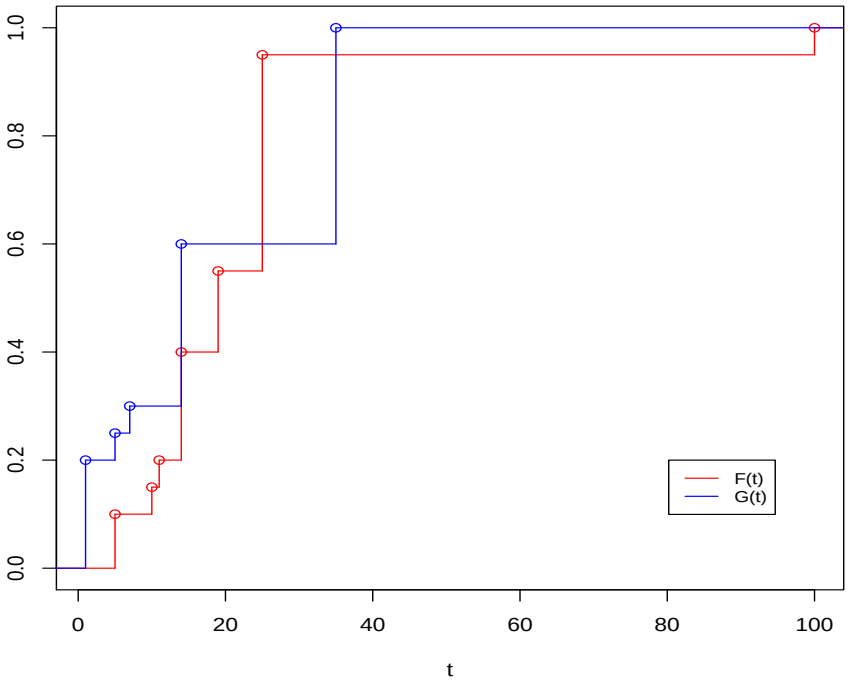


Figure 3.2: Distributions of F and G

3.2.3 $\mathbf{DE}(X, Y)$ is a pseudometric

Claim: $\mathbf{DE}(X, Y)$ is a pseudometric.
Proof: Let (X, d_X, μ_X),(Y, d_Y, μ_Y),(Z, d_Z, μ_Z) be given.

1. $\mathbf{DE}(X, X) = \frac{1}{2}(\int_{\mathbb{R}} |S_{X,1}(t) - S_{X,1}(t)|dt) = 0$
2. $\mathbf{DE}(X, Y) = \frac{1}{2}(\int_{\mathbb{R}} |S_{X,1}(t) - S_{Y,1}(t)|dt) = \frac{1}{2}(\int_{\mathbb{R}} |S_{Y,1}(t) - S_{X,1}(t)|dt) = \mathbf{DE}(Y, X)$
3. $\mathbf{DE}(X, Y) \leq \mathbf{DE}(X, Z) + \mathbf{DE}(Z, Y)$ (3.95)

To see the triangle-inequality: Denote with

$$f := S_{X,1}(t)$$
$$g := S_{Y,1}(t)$$
$$h := S_{Z,1}(t).$$

Then 3.95 reformulates to:

$$\int_{\mathbb{R}} |f - g|dt \leq \int_{\mathbb{R}} |f - h|dt + \int_{\mathbb{R}} |h - g|dt. \quad (3.96)$$

With **??** it follows $\int_{\mathbb{R}} |f-g|dt = |\int_{\mathbb{R}} f-g|dt$. Therefore it is sufficient to show:

$$|\int_{\mathbb{R}} f-g \quad dt| \leq |\int_{\mathbb{R}} f-h \quad dt| + |\int_{\mathbb{R}} h-g \quad dt| \tag{3.97}$$

$$\Leftrightarrow |\int_{\mathbb{R}} f - \int_{\mathbb{R}} g \quad dt| \leq |\int_{\mathbb{R}} f - \int_{\mathbb{R}} h \quad dt| + |\int_{\mathbb{R}} h - \int_{\mathbb{R}} g \quad dt|. \tag{3.98}$$

Further simplify notation by denoting:

$$F := \int_{\mathbb{R}} f$$
$$G := \int_{\mathbb{R}} g$$
$$H := \int_{\mathbb{R}} h.$$

Then 3.98 reformulates to:

$$|F-G| \leq |F-H| + |H-G|. \tag{3.99}$$

Proof:

1. $F \leq G \leq H$
   $|F-G| \leq |F-H| \Rightarrow |F-G| \leq |F-H| + |H-G|$
2. $F \leq H \leq H$
   $|F-G| = G-F \leq H-F+G-H = G-F$
3. $G \leq F \leq H$
   $|F-G| = F-G \leq |H-F| + F-G \leq H-F+F-G = H-G$
   $H-F+F-G = H-G \leq |H-G| \leq |H-G| + |F-H|$
4. $G \leq H \leq F$
   $|F-G| = F-GF-H+H-G = |F-H| + |H-G|$
5. $H \leq G \leq F$
   $|F-G| = F-G \leq F-G+|H-F| \leq F-H+H-G = F-H$
   $F-H+H-G = F-H \leq |F-H| \leq |F-H| + |H-G|$
6. $H \leq F \leq G$
   $|F-G| = G-F \leq |F-G| = G-F+F-H = G-H = |G-H| \leq |F-H| + |H-G|$

3.22 Example *(DE is only a pseudo-metric)*
This example illustrates that ***DE*** *is not a metric but only a pseudo-metric since it does not hold that* $\boldsymbol{DE}(\mathbb{X},\mathbb{Y}) = 0 \Rightarrow \mathbb{X} = \mathbb{Y}$. *Let* $\mathbb{X} = (X, d_X, \mu_X)$

and $\mathbb{Y} = (Y, d_Y, \mu_Y)$ *with* $n_X = 2$ $n_Y = 3$ *and* $d \in \mathbb{R}$*:*

$$\mu_X = (1, 0) \tag{3.100}$$
$$\mu_Y = (1, 0, 0) \tag{3.101}$$
$$d_X = \begin{pmatrix} d & 0 \\ 0 & d \end{pmatrix} \qquad d_Y = \begin{pmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & d \end{pmatrix} \tag{3.102}$$

Then

$$s_X(x_1) = 0 \tag{3.103}$$
$$s_X(x_2) = d \tag{3.104}$$
$$S_X(-\infty) = 0 \tag{3.105}$$
$$S_X(0) = 1 \tag{3.106}$$

$$s_Y(y_1) = 0 \tag{3.107}$$
$$s_Y(y_2) = d \tag{3.108}$$
$$s_Y(y_3) = d \tag{3.109}$$
$$S_Y(-\infty) = 0 \tag{3.110}$$
$$S_Y(0) = 1 \tag{3.111}$$

Therefore $\boldsymbol{DE}(\mathbb{X}, \mathbb{Y}) = 0$*, but by construction* $\mathbb{X} \neq \mathbb{Y}$*.*

3.3 Application of the Lower Bound

In this section the application of the **FLB** to the *animal-test-set* as done in [18] is described. Furthermore I repeat the experiment here and discuss the result.

3.3.1 Downsampling

The models in the *animal-test-set* [23] contain from 7000 to 30000 points each. Additionally the points are forming a triangle-mesh as an approximation of the surface. In [18] the presented idea is to select few characteristic points that represent each model well.

In [18] the **FLB** 3.9 was used as a measure of similarity between two given models. For this purpose only few points of the models are selected in a two-step-procedure, with the aim to retain a smaller model that represents the key characteristics of the model well. In the first step the *farthest point sampling procedure* is applied in the euclidean space and $n_{\text{euclidean}} = 4000$ points are selected.

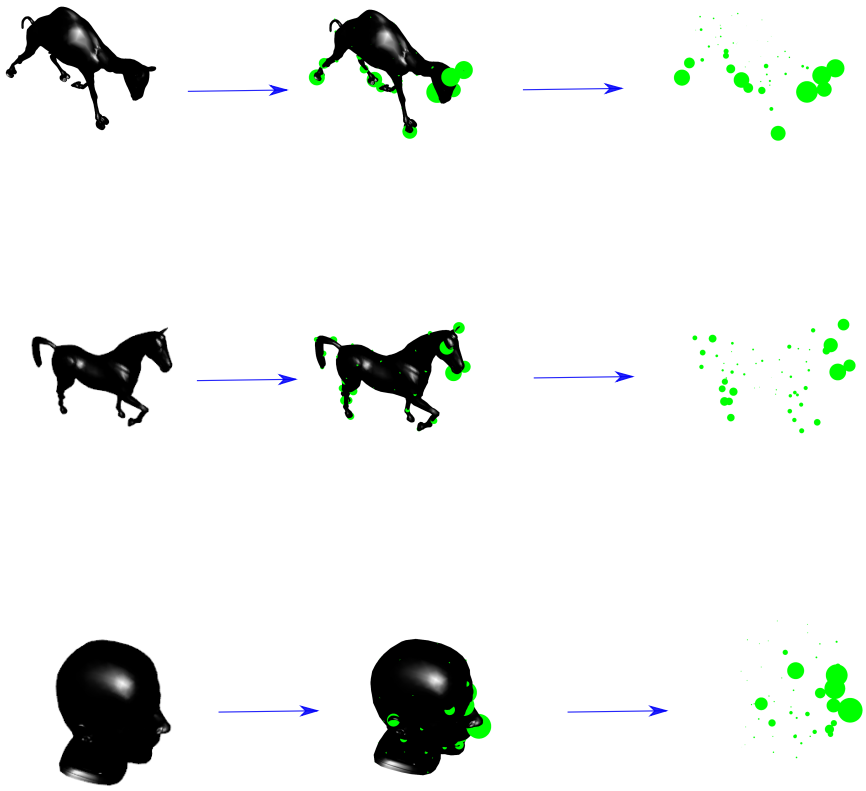


Figure 3.3: From each model a few characteristic points are selected. Displayed are the models of *camel*, *horse* and *head*.

Let $\hat{X}_k$ denote this reduced model. Then an intrinsic distance (or graph distance) was defined using Dijkstra's algorithm on the graph $G(X_k)$ with vertex set X_k, where each vertex is connected by an edge to those vertices with which it shares a triangle. Since $\hat{X}_k \subset X_k$, by restriction, one endows $\hat{X}_k$ with this intrinsic distance as well. They further sub-sampled $\hat{X}_k$, again using the farthest point procedure (with the distance computed using $G(X_k)$), and they retained only 50 points. Denote the resulting set by $\mathbb{X}_k$. They then endowed $\mathbb{X}_k$ with the *normalized* distance metric inherited from the Dijkstra procedure described above and a probability measure based on Voronoi partitions: the mass (measure) at point $x \in \mathbb{X}_k$ equals the proportion of points in $\hat{X}_k$ which are closer to x than to any other point in $\mathbb{X}_k$.

From each model X_k one thus obtains a discrete mm-space $(\mathbb{X}_k, d^{(k)}, \nu^{(k)})$. A matrix $((d_{ij}))$ is then computed with the **FLB**, such that $d_{ij} = \mathcal{D}(\mathbb{X}_i, \mathbb{X}_j)$.

The authors do not elaborate much on how the parameters $n_{\text{euclidean}}$ and n_{dijkstra} were chosen. However I assume that choosing more points, even if it might be computational feasible, does not always lead to a more precise similarity-measure. I think that a good balance between using too little points, that lead to over-generalizing the model and using too many points, that lead to over-specifying of the model, has to be found. This problem reminds of the under-and-over-fitting problem, that one often comes across in machine-learning [8]. It is also conceivable that the optimal number of points is problem-specific. That means that a good parameter for the *animal-test-set* does not necessarily transfer to the protein-test-set.

3.3.2 Nearest Neighbour Classification and Cross-Validation

Given the pairwise similarities of all models $((d_{ij}))$ the proposed classification-method in [18] is a $1-$NN-method (1-Nearest-Neighbor).

With any prediction-problem one is usually interested in how well the proposed method will work on new yet unseen data [6]. Hence one often splits the given data into a training-and test-set. An algorithm then computes a model with the training-set and the accuracy is then calculated based on the prediction-accuracy, that the model achieves on the test-set. The drawback of this approach is however, that the data available for the training is smaller. Also for small data-sets the choice of the test-set might not be representative and choosing a different test-set might lead to different results. By doing the split into a training- and test-set multiple times and averaging over the obtained performance, one gets an estimate for the actual prediction-accuracy of the model. This procedure is called cross-validation [15]. Many different variations of this approach exist. In [18] the following was done: from each class of *lions*, *cats*, *heads*, *horses*, *elephants*, *flamingos*, *camel* 1 model is chosen at random. The prediction for each chosen model is obtained from the nearest-neighbor computed with $((d_{ij}))$ of the remaining not selected models. This process is repeated 10000 times to get an estimate of how well the method works.

3.3.3 Results on the *animal-test-set*

I implemented this downsampling-procedure and classification-method in R and reused it on the *animal-test-set* for further investigation of the parameters $n_{\text{euclidean}}$ and n_{dijkstra}. Memoli uses the **FLB**, whereas I use the **DE** for the calculation of the dissimilarity as I have shown that it holds that $\mathbf{FLB} = \mathbf{DE}$ in the finite case in Theorem 3.14. Additionally attempting to reproduce the results from [18] assures that my implementation does not contain any larger deviations from the original implementation, let alone errors. Initially I used $n_{\text{euclidean}} = 4000$ and $n_{\text{dijkstra}} = 50$. In figure 3.1 the confusion-matrix obtained with the procedure described in 3.3.2 is shown.

On the given models the method seems to work extremely well, only the classes *cats* and *lions* are separated poorly. However as described in [23] these models are computationally derived from similar 3D-models, on top of the obvious fact that lions and cats share some natural similarity, and hence are expected to be difficult to distinguish.

Surprisingly the heatmap and dendrogram as shown in figure 3.3.3 do not reveal the same quality of information as the NN-method. With the dendrogram one is only capable of separating *heads* from non-*heads*.

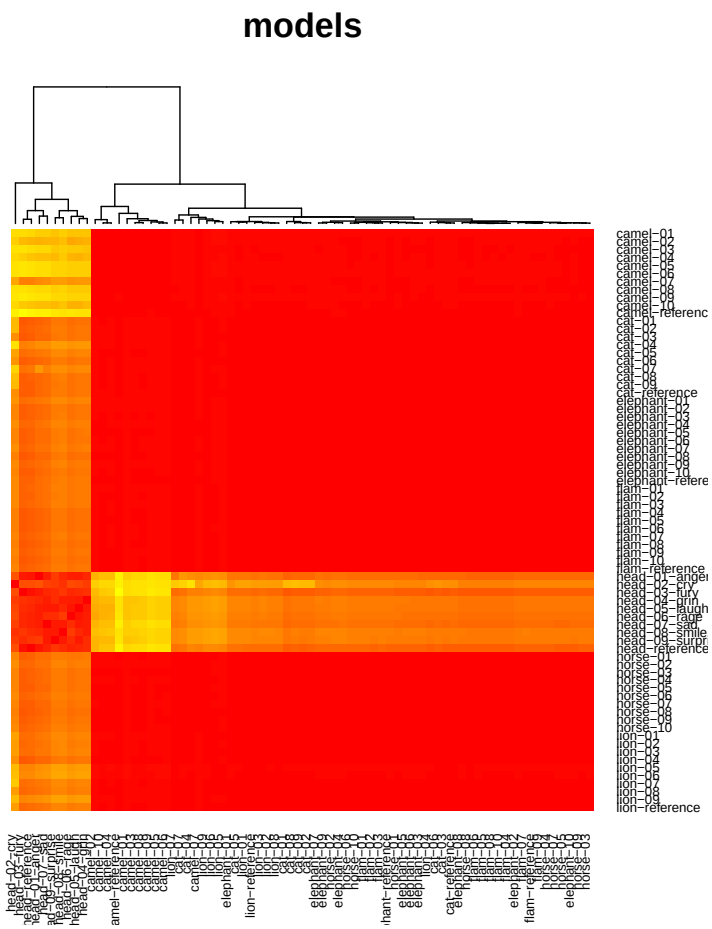


Figure 3.4: A shortest path between two points on the surface

Figure 3.5: 50 points selected from the model. Bigger points indicate a higher weight.

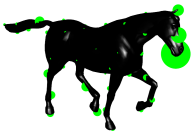


Figure 3.6: 50 points selected from the model. Bigger points indicate a higher weight.

Figure 3.7: 50 points selected from the model. Bigger points indicate a higher weight.

Table 3.1: Confusion-matrix of 1−NN-classification

	camel	**cat**	**elephant**	**flamingo**	**head**	**horse**	**lion**
camel	9153	0	0	0	0	0	847
cat	0	5130	0	102	0	905	3863
elephant	0	0	7236	914	0	1850	0
flamingo	0	0	0	10000	0	0	0
head	0	0	0	0	10000	0	0
horse	0	0	0	0	0	10000	0
lion	0	2888	0	0	0	0	7112

Chapter 4

Applying metric Geometry to Protein-isosurfaces

The theory as presented in [18] and the promising results on a somewhat comparable problem of classification of 3-dimensional objects (the *animal-test-set*) were presented in the last chapter. In this chapter it will be discussed how the **DE** can be applied in a reasonable way to retrieve a measure of dissimilarity for two given proteins.

4.1 Preprocessing - generating the Surfaces from pdb-files

Visual molecular dynamics (VMD) [13] is a molecular graphics program designed for the display and analysis of molecular structures. A pdb-file [4] (short for Protein-Data-Bank) is a standard format for proteins. For the purpose of this thesis a pdb-file can be thought of a file that stores geometrical properties which includes the relative positions of the atoms forming the protein. The pdb-files are converted to pqr-files with the program PDB2PQR [9]. A pqr-file can be thought of a pdb-file where additionally information about the electric charge of each atom is added. With the corresponding plugin of VMD with the amber-force-field [2] the electrostatic potential of the given protein is calculated. Then the native exporting-option of VMD to export this surface to a wavefront-obj-file is used. The wavefront-obj-file-format contains the surface as a triangle-mesh. The package *readobj*[14] in R is then used to read in the file. Some processing steps are then necessary to merge the triangle-mesh into one connected mesh. For this the packages *igraph*[7], $\cdots$ are used.

In addition the position of the first *cystein*-atom of the active center of the given protein is extracted with a script written in R.

All in all with the above steps completed, for each protein one is given:

- A file containing the coordinates in 3-dimensional-euclidean space of the points that form the positive potential
- A file containing the coordinates in 3-dimensional-euclidean space of the points that form the negative potential
- A file containing the coordinates in 3-dimensional-euclidean space of the position of the active center.

4.2 Calculating a Dissimilarity-measure

Given two proteins with positive and negative iso-surface, given as sets of points in 3-dimensional euclidean space, how can one calculate the dissimilarity? Here the different methods that were applied, are presented. The performance of the methods is investigated in section 6 **Results**. Following the notations in chapter 3.1:

A protein will be modeled in the following way: Given a mp-space $\mathbb{X}^{+-} := (X^{+-}, d_X^{+-}, \mu_X^{+-})$ with $X^{+-} = (X_1^{+-}, \cdots, X_{n_X^{+-}}^{+-})$ and $X^+, X^- \subseteq X^{+-}$. $X^+ = (X_1^+, \cdots, X_{n_X^+}^+)$ and $X^- = (X_1^-, \cdots, X_{n_X^-}^-)$ resemble the positive respectively negative potential of the iso-surface of the protein. Further denote $\mathbb{X}^+ := (X^+, d_X^+, \mu_X^+)$ with

$$d_X^+(x, x') := d_X^{+-}(x, x') \quad \forall x, x' \in X^+ \tag{4.1}$$

$$\mu_X^+(x) := \frac{\mu_X^{+-}(x)}{\sum\limits_{x' \in X^+} \mu_X^{+-}(x')} \quad \forall x \in X^+ \tag{4.2}$$

$$\tag{4.3}$$

and $\mathbb{X}^- := (X^-, d_X^-, \mu_X^-)$

$$d_X^-(x, x') := d_X^{+-}(x, x') \quad \forall x, x' \in X^- \tag{4.4}$$

$$\mu_X^-(x) := \frac{\mu_X^{+-}(x)}{\sum\limits_{x' \in X^-} \mu_X^{+-}(x')} \quad \forall x \in X^-. \tag{4.5}$$

Obviously by construction $\mathbb{X}^+$ and $\mathbb{X}^-$ are also mp-spaces.

4.2.1 Independently calculating for pos and neg ($\mathbf{DE}^+$, $\mathbf{DE}^-$)

The first and most straight forward approach is to calculate the dissimilarities independently for the positive and negative potentials. This will be denoted with

$$\mathbf{DE}^+(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) := \mathbf{DE}(\mathbb{X}^+, \mathbb{Y}^+) \tag{4.6}$$

respectively

$$\mathbf{DE}^-(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) := \mathbf{DE}(\mathbb{X}^-, \mathbb{Y}^-). \tag{4.7}$$

As a consequence the relation of the positive potential to the negative potential does not have an impact on the assigned similarity-value.

4.2.2 SumMethod $\mathbf{DE}^{+-}$

With this approach the relation between the positive and negative potential is modeled as follows:

$$\begin{aligned}\mathbf{DE}_c^{+-}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) := \quad & c_1 \cdot \mathbf{DE}(\mathbb{X}^+, \mathbb{Y}^+) \\ & + c_2 \cdot \mathbf{DE}(\mathbb{X}^-, \mathbb{Y}^-) \\ & + c_3 \cdot \mathbf{DE}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})\end{aligned}$$

where $c = (c_1, c_2, c_3) \in \mathbb{R}^3$.
With $c = (1, 0, 0)$ it holds that

$$\mathbf{DE}^{+-}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) = \mathbf{DE}^+(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) \tag{4.8}$$

and $c = (0, 1, 0)$ it holds that

$$\mathbf{DE}^{+-}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) = \mathbf{DE}^-(\mathbb{X}^{+-}, \mathbb{Y}^{+-}). \tag{4.9}$$

4.2.3 2d-Hist ($\mathbf{DE}_{2d}$)

In attempt to extend the **DE** to multiple features one can define:

$$s_X^+(i) = \sum_{k=1}^{n_X} \mu_X^+(k) d_X^+(i,k)$$

$$s_X^-(i) = \sum_{k=1}^{n_X} \mu_X^-(k) d_X^-(i,k)$$

$$s_X^{+-}(i) = (s_X^+(i), s_X^-(i))$$

$$t_X := \{s_X^{+-}(i) | i \in \{1, \cdots, n_X\}\}$$

t_X now simply becomes a set of pairs instead of numbers.

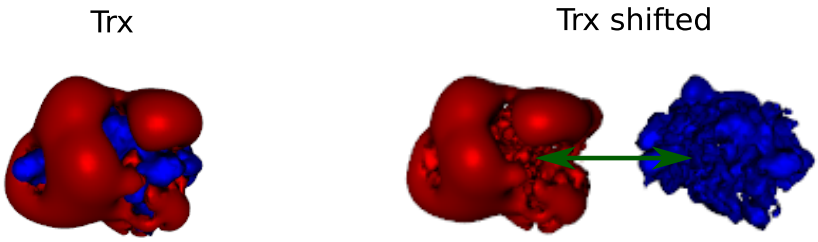


Figure 4.1: The same model, except the relative position of the positive and negative surface has changed.

$$B_{X,u} := \{i | t_X(i)_1 \leq u_1 \quad \text{and} \quad t_X(i)_2 \leq u_2, i \in \{1, \cdots, n_X\}\}$$

$$S_X(u) = \sum_{i \in B_{X,u}} \mu_X(i).$$

Essentially now S_X and S_Y are $2-$dimensional distributions. $t_i := (t_{i1}, t_{i2})$:

$$\mathbf{DE}_{2d}(\mathbb{X}, \mathbb{Y}) = \frac{1}{2} \sum_{i_1=1}^{2n-1} \sum_{i_2=1}^{2n-1} |t_{i_1+1} - t_{i_1}||t_{i_2+1} - t_{i_2}||S_X(t_i) - S_Y(t_i)|.$$

4.2.4 An artificial Example

Here an example is set up to show that the approaches 4.2.2 and 4.2.3 can from a theoretical standpoint viewed, reveal a difference in the relation of the potentials in contrast to 4.2.1. The model of a protein (in this case Trx) is taken and the positive potential is moved by 2 units along the x-axis. The two models are displayed in figure 4.1.

4.2.5 Computational Technique

In what follows the **DE** 3.13 will be used as a measure of dissimilarity for two given mp-spaces $\mathbb{X}$ and $\mathbb{Y}$:

$$\mathbf{DE}(\mathbb{X},\mathbb{Y}) := \frac{1}{2}(\int_{\mathbb{R}} |S_X(t) - S_Y(t)|dt). \tag{4.10}$$

Furthermore in what follows a fixed number for both mp-spaces will be considered $n_X = n_Y = n < \infty$. Denote with $t_X = (\pi_{X_1}(s_X(x_1)), \cdots, \pi_{X_n}(s_X(x_n)))$ where π_X is a permutation such that $t_{X_i} \leq t_{X_{i+1}}$. t_Y analogously for s_Y. Denote with $t = (\pi_1(s_X(x_1)), \cdots, \pi_n(s_X(x_n)), \pi_{n+1}(s_Y(y_1)), \cdots, \pi_{2n}(s_Y(y_n)))$ where π is a permutation such that $t_i \leq t_{i+1}$. 4.10 can then be reformulated as:

$$\mathbf{DE}(\mathbb{X},\mathbb{Y}) = \frac{1}{2}\sum_{i=1}^{2n-1} |t_{i+1} - t_i||S_X(t_i) - S_Y(t_i)| \tag{4.11}$$

with

$$S_X(u) = \sum_{i=1}^{n} \mathbb{1}_{[s_X(x_i),\infty)}(u) \tag{4.12}$$

where

$$s_X(x_i) = \sum_{k=1}^{n} d_X(x_k, x_i)\mu_X(x_k) \tag{4.13}$$

and analogously

$$S_Y(u) = \sum_{i=1}^{n} \mathbb{1}_{[s_Y(y_i),\infty)}(u) \tag{4.14}$$

where

$$s_Y(y_i) = \sum_{k=1}^{n} d_Y(y_k, y_i)\mu_Y(y_k). \tag{4.15}$$

4.3 How to deal with the high number of points?

From a practical standpoint the point-clouds are too big to just use all points with the **DE** at the same time. Therefore something has to be done before and a direct calculation with the **DE** is not possible. Here the notations are introduced to formalize the two applied approaches.

4.1 Definiton *(subset-induced-metric) Define for a given metric space* (M, d) *and a set* $A \subseteq M$ *the* subset-induced-metric $d_{|A} : A \times A \mapsto \mathbb{R}^+$

$$d_{|A}(a) := d(a) \quad \forall a \in A.$$

4.2 Definiton *(subset-induced-measure) Define for a given measurable space* $(\Omega, \mathcal{A}, \mu)$ *and a set* $X \subseteq \Omega$ *the* σ*-algebra*

$$\mathcal{A}_{|X} := \{A \cap X : A \in \mathcal{A}\} \tag{4.16}$$

which is called trace-σ-algebra*. It holds* $A \in \mathcal{A}_{|X} \Leftrightarrow A \in \mathcal{A}$ *and* $A \subset X$*. Further define the measure* $\mu_{|X}$ *on* $\mathcal{A}_{|X}$ *as:*

$$\mu_{|X}(A) := \mu(A) \quad \forall A \in \mathcal{A}_{|X}. \tag{4.17}$$

4.3 Definiton *(Multiset) Let a set* M *and a* $f : M \to \mathbb{N}_0$ *be given. The pair* (M, f) *is called* multi-set*. Denote* $x \in (M, f) :\Leftrightarrow f(x) > 0$ *and* $x \notin (M, f) :\Leftrightarrow f(x) = 0$*. Furthermore define* $f(x) := \sum_{x' \in (M,f)} \mathbb{1}_{x=x'}(x')$*. Denote with* $supp((M, f)) := \{x \in M : f(x) > 0\}$*. Denote with* $|(M, f)| := |supp((M, f))|$ *and* $|f| := \sum_{x \in M} f(x)$*.*

Let (X, d_X, μ_X) be an mp-space and (X, f) be a multi-set. Define

$$S_{\mathcal{N}}(X, d_X, \mu_X, f) := ((X, f), d_{X|\mathrm{supp}((X,f))}, \frac{\mu_{X|\mathrm{supp}((X,f))}}{\sum_{x \in \mathrm{supp}((X,f))} \mu_{X|\mathrm{supp}((X,f))}(x)}) \tag{4.18}$$

and

$$S^{\mathrm{supp}}_{\mathcal{N}}(X, d_X, \mu_X, f) := (\mathrm{supp}((X, f)), d_{X|\mathrm{supp}((X,f))}, \frac{\mu_{X|\mathrm{supp}((X,f))} \cdot f(x)}{\sum_{x \in \mathrm{supp}((X,f))} \mu_{X|\mathrm{supp}((X,f))}(x)}). \tag{4.19}$$

That means by definition of $S^{\mathrm{supp}}_{\mathcal{N}}$ if for an $x \in X$ it holds that $f(x) > 1$, or in other words, if a point is sampled more than once, the mass of this point is multiplied with the times it was sampled. The following example illustrates this.

4.4 Example *(Unite mass)*

$$\mu_X = \frac{1}{2} \cdot \begin{pmatrix} 1 & 1 \end{pmatrix} \tag{4.20}$$

$$f = \begin{pmatrix} 10 & 10 \end{pmatrix}$$

$$F = \frac{1}{2} \cdot \begin{pmatrix} 1 & 2 \end{pmatrix} \tag{4.21}$$

$$\mu_X^* = \frac{1}{2} \cdot \begin{pmatrix} 2 \end{pmatrix} \tag{4.22}$$

$$f^* = \begin{pmatrix} 20 \end{pmatrix}$$

$$F^* = \frac{1}{2} \cdot \begin{pmatrix} 2 \end{pmatrix} \tag{4.23}$$

It holds that $F(t) = F^*(t)$. *If one considers the mp-space* (X^*, d_X, μ_X) *and the multi-set* (X^*, N) *with* $N(x) = 2$ *then* $S_{\mathcal{N}}^{supp}(X^*, d_X, \mu_X^*, N) = (X, d_X, \mu_X)$.

Define $S_{\mathcal{V}}^{\text{supp}}(X, d_X, \mu_X, f) := (\text{supp}((X, f)), d_{X|\text{supp}((X,f))}, \mu_{\mathcal{V}})$ with

$$c(x) := |\{x' \in X : d_X(x, x') \leq d_X(s, x') \quad \forall s \in \text{supp}((X, f))\}|$$
$$\mu_{\mathcal{V}}(x) := \frac{c(x)}{\sum_{x' \in \text{supp}((X,f))}}.$$

4.3.1 2-step downsampling

This is the approach that was presented in [18] and described earlier in section 3.3.1. A 3-dimensional object here is modeled as an mp-space $\mathbb{X} = (X, d_X, \mu_X)$ and an additional metric space $(X, d_{X_{geo}})$ specifying the geodesic distances of the points in the model. Now one wants to select few characteristic points that resemble the model well. Therefore in two steps points are selected from the model. In the first step with the euclidean metric (d_X) and in the second step from the now smaller model points are selected with the geodesic metric $(d_{X_{geo}})$.

Given a metric space (M, d) and a $p \in M$ define the *farthest point procedure*-sequence $\text{FPS}(M, d, p)$ as follows:

$$\text{FPS}_1 := p$$
$$\text{FPS}_i := \underset{a \in M \setminus \{\text{FPS}_1, \cdots, \text{FPS}_{i-1}\}}{\arg\max} \min_{b \in \{\text{FPS}_1, \cdots, \text{FPS}_{i-1}\}} d(a, b) \quad i \in \{1, \cdots, |M|\}.$$

Given a $n_d \leq n_e \leq n_X \in \mathbb{N}$ one now does the following: Define $\forall x \in X \quad N_e(x) := \mathbb{1}_{\{\text{FPS}_{\{1,\cdots,n_e\}}(X, d_X, p)\}}(x)$ and $X_e := \text{supp}((X, N_e))$. Then define $\forall x \in X \quad N_{\text{geo}}(x) := \mathbb{1}_{\{\text{FPS}_{\{1,\cdots,n_{\text{geo}}\}}(X_e, d_{X|X_e}, p)\}}(x)$. Then $S_{\mathcal{V}}^{\text{supp}}(X, d_X, \mu_X, N_{\text{geo}})$ is the 2-step-down-sampled model. Define $\mathcal{V}(\mathbb{X}) := S_{\mathcal{V}}^{\text{supp}}(X, d_X, \mu_X, N_{\text{geo}})$.

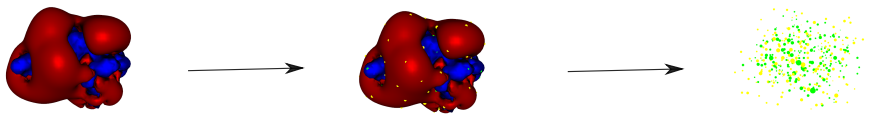


Figure 4.2: Only few points are selected

With this notation the examined models are noted as follows:

$$\mathbf{DE}^{+}(\mathcal{V}(\mathbb{X}^{+}),\mathcal{V}(\mathbb{Y}^{+})) \tag{4.24}$$
$$\mathbf{DE}^{-}(\mathcal{V}(\mathbb{X}^{-}),\mathcal{V}(\mathbb{Y}^{-})) \tag{4.25}$$
$$\mathbf{DE}_{c}^{+-}(\mathcal{V}(\mathbb{X}^{+})\bigcup\mathcal{V}(\mathbb{X}^{-}),\mathcal{V}(\mathbb{Y}^{+})\bigcup\mathcal{V}(\mathbb{Y}^{-})) \tag{4.26}$$
$$\mathbf{DE}_{2d}(\mathcal{V}(\mathbb{X}^{+})\bigcup\mathcal{V}(\mathbb{X}^{-}),\mathcal{V}(\mathbb{Y}^{+})\bigcup\mathcal{V}(\mathbb{Y}^{-})). \tag{4.27}$$

To simplify notation I introduce the following abbreviations:

$$\mathbb{V}_{\mathbf{DE}^{+}}(\mathbb{X}^{+-},\mathbb{Y}^{+-}) \quad :=\mathbf{DE}^{+}(\mathcal{V}(\mathbb{X}^{+}),\mathcal{V}(\mathbb{Y}^{+})) \tag{4.28}$$
$$\mathbb{V}_{\mathbf{DE}^{-}}(\mathbb{X}^{+-},\mathbb{Y}^{+-}) \quad :=\mathbf{DE}^{-}(\mathcal{V}(\mathbb{X}^{-}),\mathcal{V}(\mathbb{Y}^{-})) \tag{4.29}$$
$$\mathbb{V}_{\mathbf{DE}_{c}^{+-}}(\mathbb{X}^{+-},\mathbb{Y}^{+-}) \quad :=\mathbf{DE}_{c}^{+-}(\mathcal{V}(\mathbb{X}^{+})\bigcup\mathcal{V}(\mathbb{X}^{-}),\mathcal{V}(\mathbb{Y}^{+})\bigcup\mathcal{V}(\mathbb{Y}^{-})) \tag{4.30}$$
$$\mathbb{V}_{\mathbf{DE}_{2d}}(\mathbb{X}^{+-},\mathbb{Y}^{+-}) \quad :=\mathbf{DE}_{2d}(\mathcal{V}(\mathbb{X}^{+})\bigcup\mathcal{V}(\mathbb{X}^{-}),\mathcal{V}(\mathbb{Y}^{+})\bigcup\mathcal{V}(\mathbb{Y}^{-})). \tag{4.31}$$

4.3.2 Repeated Sub-sampling

Felix used the **DE** as described in section 3 in his master-thesis [3]. Since one can not compute the **DE** for all points in X and Y at once, since (n_X, n_Y is too large), he performed a sampling procedure. His idea was to sample uniformly n points of both point-clouds. Then calculate the **DE** for the two samples. This is repeated for m times. The values for each sampling-step are stored in a vector. A histogram is then built from these values. This comparison is done once with X, X and X, Y. Then the **emd** is calculated as the final value between the two histograms. Felix used a uniform-distribution for the measures $\mu_X = (\frac{1}{n_X}, \cdots, \frac{1}{n_X}))$ $\mu_Y = (\frac{1}{n_Y}, \cdots, \frac{1}{n_Y}))$.

Here I will try to describe this approach in a more formal way with the modifications that I added to the approach.

The setting is the following: Given two mp-spaces

$$\mathbb{X} = (X, d_X, \mu_X) \tag{4.32}$$
$$\mathbb{Y} = (Y, d_Y, \mu_Y) \tag{4.33}$$

and $n, m \in \mathbb{N}$ a distance should be calculated. Let $U_n : \mathbb{N} \to \mathbb{R}^+$

$$U_n(u) = \begin{cases} \frac{1}{n}, & \text{if } 1 \leq u \leq n \\ 0, & \text{otherwise} \end{cases} \tag{4.34}$$

Let Z_X a random variable with density $U_{n_X}(u)$.

Let $s_{i,j} \sim Z_X$ i.i.d. for $j \in \{1, \cdots, n\}$ and for $i \in \{1, \cdots, m\}$. Define $f_i(x) := \sum_{j=1}^{m} \mathbb{1}_{\{s_{i,j}\}}(x)$.

Denote with $R(\mathbb{X}, n, m) := (R_1, \cdots, R_m)$

$$R_i := (S_{\mathcal{N}}^{\text{supp}}(X, d_X, \mu_X, f_i)). \tag{4.35}$$

That means $R(\mathbb{X}, n, m)$ is a random-variable.

Let $R_X, R'_X \sim R(\mathbb{X}, n, m)$.

Denote with $Q^m(R_X, R'_X) = (Q_1, \cdots, Q_m)$

$$Q^m(R_X, R'_X)_i := \mathbf{DE}(R_{X_i}, R'_{X_i}). \tag{4.36}$$

Let $R_Y, R'_Y \sim R(\mathbb{Y}, n, m)$.

Denote

$$\mathbb{S}_{\mathbf{DE}}(\mathbb{X}, \mathbb{Y}) := \quad \mathbf{emd}(Q^m(R_X, R'_X), Q^m(R_X, R_Y)) \tag{4.37}$$

$$+\mathbf{emd}(Q^m(R_Y, R'_Y), Q^m(R_X, R_Y)). \tag{4.38}$$

While Felix originally calculates the final value as

$$\mathbb{S}_{\mathbf{DE}}(\mathbb{X}, \mathbb{Y})' := \mathbf{emd}(Q^m(R_X, R'_X), Q^m(R_X, R_Y)) \tag{4.39}$$

I argue that by adding the second term $\mathbb{S}_{\mathbf{DE}}(\mathbb{X}, \mathbb{Y})$ becomes symmetric.

A desirable property of $\mathbb{S}_{\mathbf{DE}}$ would be the triangle-in-equality. Assume that **emd** is a metric. Denote for a given n with

$$a := Q^\infty(X, X) := \lim_{m \to \infty} Q^m(X, X) \tag{4.40}$$

$$b := Q^\infty(X, Y) \tag{4.41}$$

$$c := Q^\infty(Y, Y) \tag{4.42}$$

$$d := Q^\infty(X, Z) \tag{4.43}$$

$$e := Q^\infty(Z, Z) \tag{4.44}$$

$$f := Q^\infty(Z, Y) \tag{4.45}$$

Then for the triangle-in-equality of $\mathbb{S}_{\mathbf{DE}}$ it would have to hold for $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ that

$$\mathbb{S}_{\mathbf{DE}}(\mathbb{X}, \mathbb{Y}) \leq \mathbb{S}_{\mathbf{DE}}(\mathbb{X}, \mathbb{Z}) + \mathbb{S}_{\mathbf{DE}}(\mathbb{Z}, \mathbb{Y}) \tag{4.46}$$

$$\tag{4.47}$$

which translates to

$$\mathbf{emd}(a,b) + \mathbf{emd}(c,b) \leq \mathbf{emd}(a,d) + \mathbf{emd}(e,d) + \mathbf{emd}(e,f) + \mathbf{emd}(c,f). \tag{4.48}$$

This however does not hold. Fix a, c, d, e, f. Then b can freely be chosen to make $\mathbf{emd}(a,b) + \mathbf{emd}(c,b)$ arbitrarily large. The right-hand-side of the equation does not include b however.

With this notation the examined models are noted as follows:

$$\mathbb{S}_{\mathbf{DE}^+}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) \tag{4.49}$$
$$\mathbb{S}_{\mathbf{DE}^-}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) \tag{4.50}$$
$$\mathbb{S}_{\mathbf{DE}_c^{+-}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}) \tag{4.51}$$
$$\mathbb{S}_{\mathbf{DE}_{2d}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}). \tag{4.52}$$

4.4 The Active Site

The active site is of particular interest in many protein-protein-reactions.

In order to model the importance of the active site as opposed to some long-range interactions in the iso-surface, the following two approaches have been tested. One approach is using only the k nearest neighbors to the active center of a given Protein. Another approach is to change the measures μ_X and μ_Y depending on the distance to the active center. Both approaches are examined in section 6.

4.4.1 k nearest neighbors to the active Site

Let (X, d_X, μ_X) a mm-space and $a \in \mathbb{R}^3$ the active center and $k < n_X$. Denote with $d_a = d_{a1}, \cdots, d_{aX_n}$ the distances of all points to the active center $d_{ai} := ||x_i - a||$. Furthermore let $d_{ai} \leq d_{ai+1} \quad \forall i \in \{1, \cdots, X_n - 1\}$ (that means we permute X). Then denote with (X_a, d_{Xa}, μ_{Xa}) the reduced model with the following properties:

$$X_a := \{x_1, \cdots, x_k\} \tag{4.53}$$
$$d_{X_a i,j} = ||x_i - x_j|| \quad x_i, x_j \in X_a \tag{4.54}$$
$$\mu_{Xa} = (\mu_{Xa1}, \cdots, \mu_{Xak}) \tag{4.55}$$
$$\mu_{Xai} := \frac{\mu_{Xi}}{\sum_{j=1}^{k} \mu_{Xaj}}. \tag{4.56}$$

4.4.2 A measure based on the distance to the active Site

Let $a \in \mathbb{R}^3$ be the active center. Each point gets a weight according to:

$$\begin{aligned} c &:= \min_{x \in X} ||X(i) - a|| \\ f &:= \max_{x \in X} ||X(i) - a|| \\ m &:= \frac{1-w}{f-c} \\ \mu_X(i) &:= \frac{w + m \cdot (||X(i) - a|| - c)}{N} \end{aligned} \tag{4.57}$$

where N is a normalizing constant. With w the factor between the closest and furthest point is specified. With $w = 1$ a uniform distribution is modeled. This is done for both the negative and positive potential independently. That means the calculation is:

$$\begin{aligned} c^+ &:= \min_{x \in X^+} ||X^+(i) - a|| \\ f^+ &:= \max_{x \in X^+} ||X^+(i) - a|| \\ m^+ &:= \frac{1-w^+}{f^+ - c^+} \\ c^- &:= \min_{x \in X^-} ||X^-(i) - a|| \\ f^- &:= \max_{x \in X^-} ||X^-(i) - a|| \\ m^- &:= \frac{1-w^-}{f^- - c^-} \\ \mu_X^+(i) &:= \frac{w^+ + m^+ \cdot (||X^+(i) - a|| - c^+)}{N^+} \\ \mu_X^-(i) &:= \frac{w^- + m^- \cdot (||X^-(i) - a|| - c^-)}{N^-}. \end{aligned}$$

Chapter 5

Classification

5.1 The PAC learning model [19, Section 2.1]

In this section the basics of machine-learning are presented. The structure and notation is taken from [19, Foundations of machine-learning]. Small adjustments are made in an attempt to reduce the complexity of the formalism to fit closer to the protein-problem.

Denote by $\mathcal{X}$ the set of all possible *examples* or *instances*. The set of all possible *labels* is denoted by $\mathcal{Y}$. A *concept* $c : \mathcal{X} \to \mathcal{Y}$ is a mapping from $\mathcal{X}$ to $\mathcal{Y}$. A *concept class* is a set of concepts one may wish to learn and is denoted by C. Assume that examples are independently and identically distributed (i.i.d.) according to some fixed but unknown distribution D. The learning problem is then formulated as follows. The learner considers a fixed set of possible concepts H called a *hypothesis set*, which may not coincide with C. He receives a sample $S = (x_1, \cdots, x_m)$ drawn i.i.d. according to D as well as the labels $(c(x_1), \cdots, c(x_m))$, which are based on a specific target concept $c \in C$ to learn. His task is to use the labeled sample S to select a hypothesis $h_S \in H$ that has a small *generalization error* with respect to the concept c.

5.1 Definiton *[19, Def. 2.1]*
Given a hypothesis $h \in H$, a target concept $c \in C$, and an underlying distribution D, the generalization error *or* risk *of h is defined by*

$$R(h) := \Pr_{x \sim D}[h(x) \neq c(x)] = \mathop{E}_{x \sim D}[1_{h(x) \neq c(x)}] \tag{5.1}$$

where 1_ω is the indicator function of the event ω.

5.2 Definiton *[19, Def. 2.2]*
Given a hypothesis $h \in H$, a target concept $c \in C$, and a sample $S = (x_1, \cdots, x_m)$ the empirical error *or* empirical risk *of h is defined by*

$$\hat{R}(h) := \frac{1}{m}\sum_{i=1}^{m} 1_{h(x_i)\neq c(x_i)}. \tag{5.2}$$

5.3 Definiton *[19, Def. 2.3]*
A concept class C is said to be PAC-learnable *if there exists an algorithm $\mathcal{A}$ and a polynomial function poly$(\cdot,\cdot,\cdot,\cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions D on $\mathcal{X}$ and for any target concept $c \in C$, the following holds for any sample size $m \geq$ poly$(\frac{1}{\epsilon}, \frac{1}{\delta}, n,$ size$(c))$:*

$$\Pr_{S\sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta. \tag{5.3}$$

If $\mathcal{A}$ further runs in poly$(\frac{1}{\epsilon}, \frac{1}{\delta}, n,$ size$(c))$, then C is said to be efficiently PAC-learnable. When such an algorithm $\mathcal{A}$ exists, it is called a PAC-learning algorithm for C.

5.4 Definiton *[19, Def. 4.1]*
Let h_S denote the hypothesis returned by a learning algorithm $\mathcal{A}$, when trained on a fixed sample S. Then, the leave-one-out error *of $\mathcal{A}$ on a sample S of size m is defined by*

$$\hat{R}_{LOO}(\mathcal{A}) := \frac{1}{m}\sum_{i=1}^{m} 1_{h_{S-\{x_i\}}}(x_i) \neq y_i. \tag{5.4}$$

5.5 Lemma *[19, Lemma. 4.1]*
The average leave-one-out error for samples of size $m \geq 2$ is an unbiased estimate of the average generalization error for samples of size $m-1$:

$$\underset{S\sim D^m}{E}[\hat{R}_{LOO}(\mathcal{A})] = \underset{S'\sim D^{m-1}}{E}[R(h_{S'})], \tag{5.5}$$

where D denotes the distribution according to which points are drawn.

5.1.1 Applying the PAC-formalism

Here we apply the PAC-model to our problem. $\mathcal{X}$ is the set of all proteins that exist in nature. Given one specific protein $x \in \mathcal{X}$ that is functional in a specific chemical reaction, the concept c we want to learn is which other proteins are functional in that reaction.

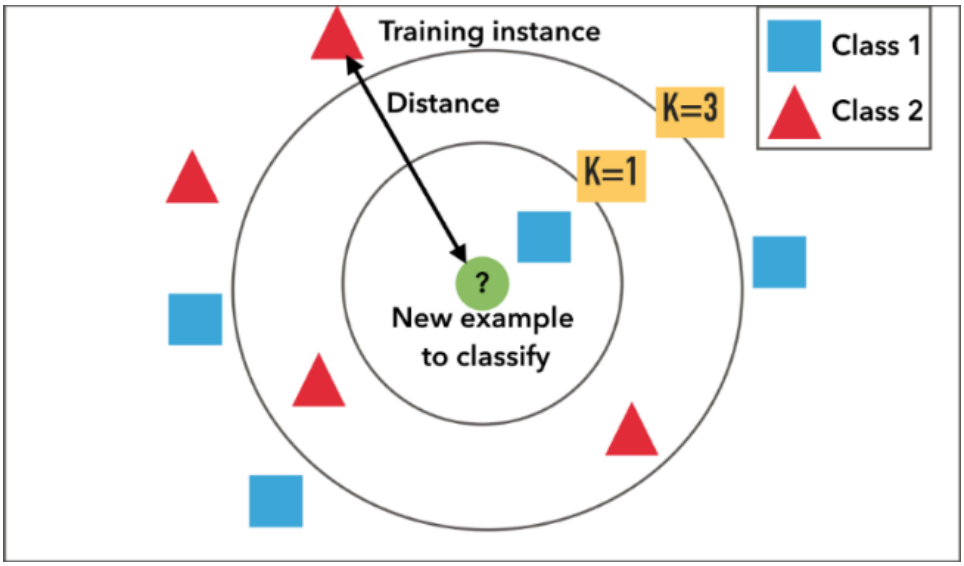


Figure 5.1: An example of the k-NN-classifier from google-images

5.2 K-Nearest-Neighbours

The k-nearest-neighbors algorithm (k-NN) is a non-parametric method [1] that can be used for classification. For each test-example the k nearest neighbors in the training-set are taken into account to make a prediction. For this purpose different approaches suitable for different settings exist.

In case of an unbalanced class-design a weighted k-NN can lead to a more accurate classifier. Weighted k-NN makes a majority-vote just as in a normal k-NN. The difference is that each class gets a weight according to the inverse of its occurence in the data.

The scenario can be modeled as follows. Given a distance-matrix $d \in \mathbb{R}^{m\times m}$ and a vector of labels $y \in \{0,1\}^m$ and a split $S_{\text{train}} \subset \{1,\cdots,m\}$, $S_{\text{test}} = \{1,\cdots,m\} \setminus S_{\text{train}}$. $y_{S_{\text{train}}}$ is known and $y_{S_{\text{test}}}$ should be predicted with a method.

Given a distance-matrix $d \in \mathbb{R}^{m\times m}$ and a $i \in S_{\text{test}}$ denote with

$$\pi^i = (\pi^i_1,\cdots,\pi^i_{|S_{\text{train}}|}) \quad \text{with} \tag{5.6}$$

$$d_{i,\pi^i_j} \leq d_{i,\pi^i_{j+1}} \tag{5.7}$$

a reordering of the indices in S_{train}. Denote with

$$y_f(0) := \frac{\sum_{j \in S_{\text{train}}} \mathbb{1}_{\{0\}}(y_{\pi_j^i})}{|S_{\text{train}}|} \tag{5.8}$$

$$y_f(1) := \frac{\sum_{j \in S_{\text{train}}} \mathbb{1}_{\{1\}}(y_{\pi_j^i})}{|S_{\text{train}}|}. \tag{5.9}$$

5.6 Definiton *For $i \in S_{test}$ define*

$$C_1(i) := y_{\underset{j \in S_{train}}{\arg\min} d(i,j)}. \tag{5.10}$$

C_1 *is called* 1-Nearest-Neighbor-Classifier.

5.7 Definiton
For $i \in S_{test}$ define

$$C_k(i) = \begin{cases} 1, & \text{if } \sum_{j \in S_{train}} \mathbb{1}_{\{1\}}(y_{\pi_j^i}) \frac{1}{y_f(1)} > \sum_{j \in S_{train}} \mathbb{1}_{\{0\}}(y_{\pi_j^i}) \frac{1}{y_f(0)} \\ 0, & \text{otherwise} \end{cases} \tag{5.11}$$

C_k *is called* weighted k-Nearest-Neighbor-Classifier.

5.2.1 Cross-Validation

The split $S_{\text{train}}, S_{\text{test}}$ is randomly generated multiple times. Each time the prediction-accuracy is evaluated with C_k. The average prediction-accuracy then is calculated.

5.2.2 Classification-Performance

Introducing the following abbreviations. Given $k \in \mathbb{R}$ and S_{train}, S_{test} and $d \in \mathbb{R}^m \times \mathbb{R}^m$ define:

$$\mathbf{TP} := |\{i \in S_{test} : C_{k,S_{train}}(x_i) = y_i \quad \text{and} \quad y_i = 1\}|$$

$$\mathbf{TN} := |\{i \in S_{test} : C_{k,S_{train}}(x_i) = y_i \quad \text{and} \quad y_i = 0\}|$$

$$\mathbf{FP} := |\{i \in S_{test} : C_{k,S_{train}}(x_i) \neq y_i \quad \text{and} \quad y_i = 0\}|$$

$$\mathbf{FN} := |\{i \in S_{test} : C_{k,S_{train}}(x_i) \neq y_i \quad \text{and} \quad y_i = 1\}|,$$

where **TP**, **TN**, **FP**, **FN** stands for *True-positive*, *True-negative*, *False-positive*, *False-negative* respectively.

$$\mathbf{ACC} := \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}.$$

$$\mathbf{PPV} := \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}.$$

$$\mathbf{TPR} := \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}.$$

$$F_1 := 2 \cdot \frac{\mathbf{PPV} \cdot \mathbf{TPR}}{\mathbf{PPV} + \mathbf{TPR}},$$

where **ACC**, **PPV**, **TPR**, F_1 stands for *accuracy*, *positive-predicted-value*, F_1*-score*, respectively. The F_1-score is a more reliable measure of accuracy, than **ACC** when the classes are imbalanced. In table 5.1 an example is shown with an unbalanced class-distribution. $\mathbf{ACC} = \frac{30}{300} = 0.9$ even though not a single instance of the labels with $y_i = 1$ is correctly predicted. A naive estimator with $\hat{y}_i = 0 \quad \forall i$ leads to this accuracy. In table 5.3 the accuracy calculates as $\mathbf{ACC} = \frac{30+30+0+0}{30+30+240} = 0.9$. The F_1-scores of both examples calculate as $F_1 = 0$ and $F_1 = 2 \cdot \frac{\frac{30}{30+30} \cdot \frac{30}{30}}{\frac{30}{30+30} + \frac{30}{30}} = 2 \cdot \frac{\frac{1}{2} \cdot 1}{\frac{1}{2} + 1} = \frac{2}{3}$. That means in this case maximizing the F_1-score would lead to preferring the predictions that produced the confusion-matrix in table 5.3.

Table 5.1: Two Confusion-matrices

Table 5.2

	$y_i = 1$	$y_i = 0$
$\hat{y}_i = 1$	0	0
$\hat{y}_i = 0$	30	270

Table 5.3

	$y_i = 1$	$y_i = 0$
$\hat{y}_i = 1$	30	30
$\hat{y}_i = 0$	0	240

Chapter 6

Results

In this section the different models presented in chapter 4 are tested. As classification-method the C_k (the k-weighted-nearest-neighbors as defined in 5.7) is chosen. As the data-set to train and test the performance the *106 Redoxins* are chosen. In order to get an estimate for the performance of the methods leave-one-out-cross-validation together with the F_1-score was performed. That means the general execution looks like this:

- Choose a model
- Choose parameters (k, and for $\mathbf{DE}_c^{+-}$ choose c)
- Calculate all pairwise similarity-measures
- Calculate the F_1-score.

In other words

$$\max_{c,k,\text{Model}} F_1 \to \max .$$

6.0.1 2-step-Downsampling

This includes the models:

$$\mathbb{V}_{\mathbf{DE}^+}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$$
$$\mathbb{V}_{\mathbf{DE}^-}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$$
$$\mathbb{V}_{\mathbf{DE}_c^{+-}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$$
$$\mathbb{V}_{\mathbf{DE}_{2d}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}).$$

For each of these models the parameters were set as $n_{\text{euclidean}} := 4000$ and $n_{\text{dijkstra}} \in \{2, \cdots, 4000\}$. For $\mathbb{V}_{\mathbf{DE}_c^{+-}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$ a grid-search was performed to find the optimal parameters c_1, c_2, c_3.

6.0.2 Repeated Sub-sampling

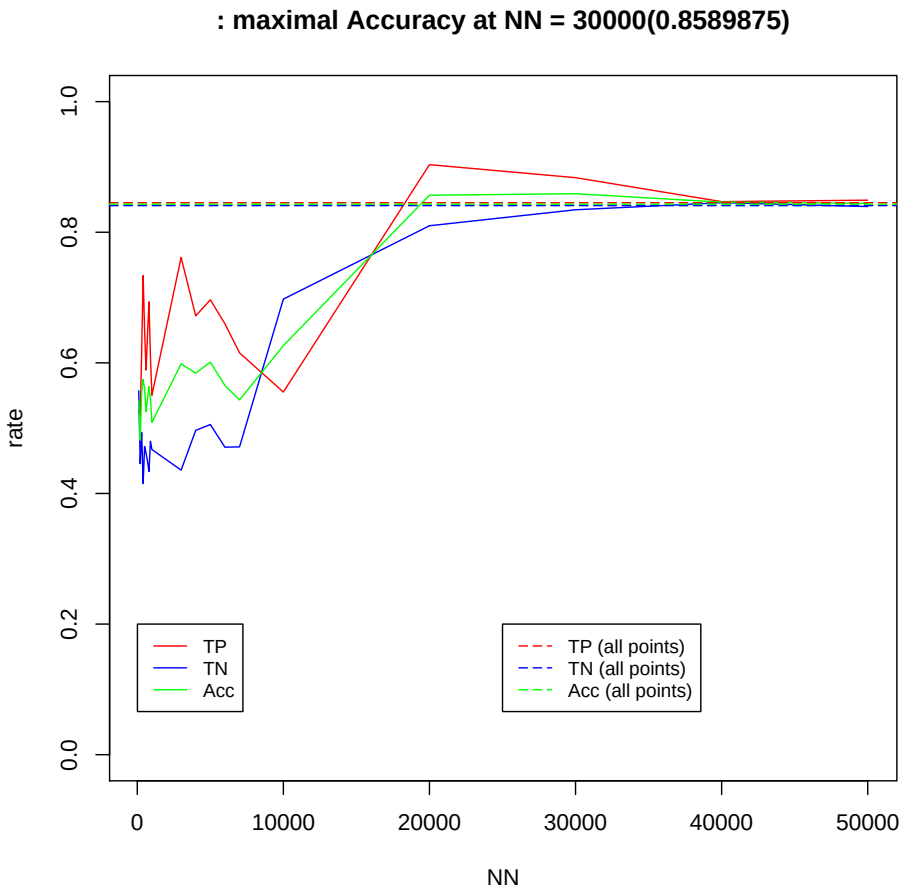
This includes the models:

$$\mathbb{S}_{\mathbf{DE}^+}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$$
$$\mathbb{S}_{\mathbf{DE}^-}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$$
$$\mathbb{S}_{\mathbf{DE}_c^{+-}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$$
$$\mathbb{S}_{\mathbf{DE}_{2d}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-}).$$

For each of these models the parameters were set as $n = 100$ and $m = 5000$. For $\mathbb{S}_{\mathbf{DE}_c^{+-}}(\mathbb{X}^{+-}, \mathbb{Y}^{+-})$ a grid-search was performed to find the optimal parameters c_1, c_2, c_3.

6.1 Active Site

As stated above, an interesting topic is the role of the active site. In order to examine the relevance for predicting the protein-interaction the *repeated-sampling-method* with parameters $n = 100, m = 500$ was tested with the reduced models $(X_a, d_{X\,a}, \mu_{X\,a})$ as described in section 4.4.1. That means for a comparison of two proteins only the closest points in relation to the active center are considered. Then the prediction-accuracy is estimated with C_k. This was done 10 times and the average prediction-accuracy is shown in figure 6.1. The dashed lines show the prediction-performance that is achieved when taking all points. The graph shows the tendency that generally taking more points leads to a higher accuracy.



6.2 Testing the model on an additional Set of Proteins

In Lilligs work-group 19 additional proteins were synthesized that are fairly similar to Thrx and the functionality was predicted with a model trained on the *106-Redoxins*-data-set. 18 of the 19 proteins were predicted as functional while all 19 proteins were functional.

6.3 Conclusion

I automatized the visual approach proposed by Lillig et. al. making it possible to be applied on a larger scale. This program is publicly available and runs under the gnu-public-license. Other biologists can also make use of it. In drug-design querying a library for candidates with this computationally

cheap technique can also be useful.

The previously stated hypothesis, that the geometrical similarity of protein-iso-surfaces can to an extend be used for predicting functionality, was confirmed. Furthermore it was confirmed that the dominating factor here lies in the positive-potential of the protein, which was proposed in [5]. This was possible by only examining the two iso-surface-values -1 and $+1$. Other additional potentials might reveal even more information.

Furthermore more evidence was collected that suggests that similarity of iso-surfaces in close proximity to the active site seems to be less important than similarity of the iso-surfaces as a whole.

Chapter 7

Appendix

7.0.1 Automatization of the visual Approach

In this chapter we want to describe the workflow of the visual approach and how we automatized time-consuming steps in it.

The basic setting is the following: given the primary-structure of a protein X and a set of proteins Y_i (with their primary-structures), we want to evaluate for each Y_i the similarity to X. In order to do so, we use VMD a visualization-tool in order to get a 3-dimensional graphical representation of the iso-surfaces of the proteins as well as of the primary-structure and the secondary-structure. We center the graphical-representation on the active center of the protein in a beforehand manually set distance. The distance to be set, depends on the class of proteins that are examined and needs expert-knowledge. Larger molecules should be looked at from a further distance than smaller molecules. The key-point is that the distance is similar for all proteins that are investigated. Then a snapshot, meaning a 2-dimensional image of what is to be seen on the screen is taken. This is done for each protein. All the images of the iso-surfaces are compiled manually and put together in a big sheet. The biological expert then examines the images and decides based on the similarity of the images of X and Y_i, if the proteins can be classified as similar or not similar. *VMD* (Visualize molecular Dynamics) is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting.

In our case the pqr-files of the proteins are needed as Input. APBS solves the equations of continuum electrostatics for large biomolecular assemblages. The output of the APBS-calculation is saved in dx-files. A dx-file specifies a voxel-grid with a certain potential-value at each grid-point. E.g. a cubic voxel-grid of size 128 contains $128^3 = 2097152$ different points. At each of these points the potential is stored. Given a certain potential VMD selects the points and interpolates between them with the marching-cubes algorithm to create a visual representation of the iso-surface. This visual rep-

resentation can be exported as wave-front-obj-file. This file then contains the surface as a triangle-mesh.

In order to extract the active center we used the pdb-files as input. Additionally we are given a list of motives that form active centers in the class of the redoxins. In our testset with 106 redoxins we are using a list that contains approximately 50 different motives, each with a length of 4 bases. The pdb-file is searched for each of the motives and outputs a warning if multiple motives are found. The corresponding position of the bases that form the active center is then used to calculate the geometric center of the bases. This geometric center is defined as location of the active center in all further methods. Note that since we only take the geometric center, we loose the information of the orientation of the active center. A script that is capable of doing this step can be found at **https://github.com/WillyBruhn/centerSelect.git**.

As a preliminary step for generating the data that was needed for the here presented thesis and [Felix]' thesis I implemented a shell-script that automatizes the above steps.

The proteins are given in pdb(protein-data-base-format). We use the command-line-tool *pdb2pqr* to convert the pdb-files to pqr-files. The pqr-file-format allows users to add charge and radius parameters to existing pdb data. This can be thought of as putting the pdbs in a water-simulation (???). Then an apbs-run is started that calculates the forces of the different atoms in the molecule. With the information added with the apbs-run, the different iso-surface values are added as graphical representation. The iso-surface values -1 and 1 were chosen. -1 in red and 1 as blue. Then the camera is positioned accordingly so that the focus is on the position of the active center in a fixed distance.

Additionally to the iso-surface-representation the primary-structure and secondary-structure is plotted. These three images per protein are then compiled into one larger overview as can be seen in figure 7.1

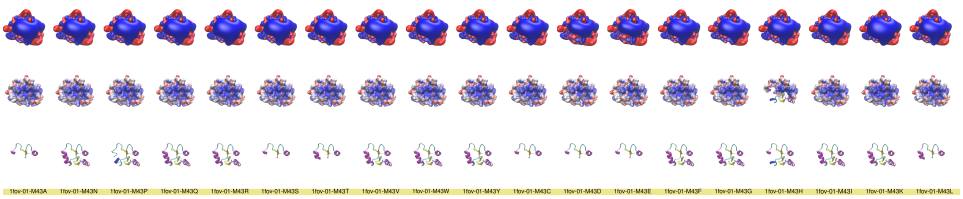


Figure 7.1: Output of MutComp - A compiled image of different proteins

Additionally I implemented a GUI for easier usage as can be seen in figure 7.2. The different parameters can here be set easily and saved for different data-sets. This was done with C++ and the Qt-framework [10].

The implementation can be found at **https://github.com/WillyBruhn/MutComp.git**.

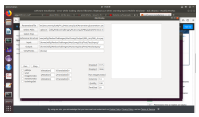


Figure 7.2: The GUI of MutComp - different parameters can be set here

7.0.2 Downloading pdbs automatically

In order to test the implementation on additional data I have written a *python*-script to download pdb-files automatically. As Input the name of a protein-target available at *https://www.drugbank.ca* has to be specified. A list of possible proteins that can interact with a specified drug from *https://www.drugbank.ca* are downloaded. Then the pdb-files are downloaded from *https://www.rcsb.org/*, if available. A file with the labels is then generated, where each pdb is assigned as functionality the target-protein according to *https://www.drugbank.ca*.

7.0.3 Predicting Protein Interactions

The script *PredictingProteinInteractions.R* wraps all the above functionalities in one script. As input a directory of pdb-files and a output-directory has to be specified. Then MutComp is executed. Then one of the mathematical model calculates all pairwise distances of all proteins. Then the k-nearest-neighbors-classification is executed.

Bibliography

[1] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3 (1992), pp. 175–185 (cit. on p. 46).

[2] Nathan A Baker et al. "Electrostatics of nanosystems: application to microtubules and the ribosome". In: *Proceedings of the National Academy of Sciences* 98.18 (2001), pp. 10037–10041 (cit. on p. 33).

[3] Felix Berens. "Quantitative comparison of protein isosurfaces with approximated Gromov-Wasserstein-distance". MA thesis. Germany: University of Greifswald, 2019 (cit. on pp. 9, 40).

[4] Helen M Berman et al. "The protein data bank". In: *Nucleic acids research* 28.1 (2000), pp. 235–242 (cit. on pp. 4, 7, 33).

[5] Carsten Berndt, Jens-Dirk Schwenn, and Christopher Horst Lillig. "The specificity of thioredoxins and glutaredoxins is determined by electrostatic and geometric complementarity". In: *Chem. Sci.* 6 (12 2015), pp. 7049–7058. URL: http://dx.doi.org/10.1039/C5SC01501D (cit. on pp. 3, 6, 52).

[6] Gavin C Cawley and Nicola LC Talbot. "On over-fitting in model selection and subsequent selection bias in performance evaluation". In: *Journal of Machine Learning Research* 11.Jul (2010), pp. 2079–2107 (cit. on p. 29).

[7] Gabor Csardi and Tamas Nepusz. "The igraph software package for complex network research". In: *InterJournal* Complex Systems (2006), p. 1695. URL: http://igraph.org (cit. on p. 33).

[8] Tom Dietterich. "Overfitting and undercomputing in machine learning". In: *ACM computing surveys* 27.3 (1995), pp. 326–327 (cit. on p. 29).

[9] Todd J Dolinsky et al. "PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations". In: *Nucleic acids research* 35.suppl_2 (2007), W522–W525 (cit. on p. 33).

[10] Qt Development Frameworks. *Qt*. 1995. URL: https://www.qt.io/ (visited on 04/09/2019) (cit. on p. 54).

[11] Steven Gold et al. "New algorithms for 2D and 3D point matching: Pose estimation and correspondence". In: *Pattern recognition* 31.8 (1998), pp. 1019–1031 (cit. on p. 3).

[12] Berthold K. P. Horn. "Closed-form solution of absolute orientation using unit quaternions". In: *J. Opt. Soc. Am. A* 4.4 (Apr. 1987), pp. 629–642. URL: http://josaa.osa.org/abstract.cfm?URI=josaa-4-4-629 (cit. on p. 3).

[13] William Humphrey, Andrew Dalke, and Klaus Schulten. "VMD: visual molecular dynamics". In: *Journal of molecular graphics* 14.1 (1996), pp. 33–38 (cit. on p. 33).

[14] Gregory Jefferis and Syoyo Fujita. *readobj: Fast Reader for 'Wavefront' OBJ 3D Scene Files*. R package version 0.3.2. 2019. URL: https://CRAN.R-project.org/package=readobj (cit. on p. 33).

[15] Ron Kohavi et al. "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *Ijcai*. Vol. 14. 2. Montreal, Canada. 1995, pp. 1137–1145 (cit. on p. 29).

[16] Walter A Koppensteiner et al. "Characterization of novel proteins based on known protein structures". In: *Journal of molecular biology* 296.4 (2000), pp. 1139–1152 (cit. on pp. 3, 6).

[17] Daniel E Koshland Jr. "The key–lock theory and the induced fit theory". In: *Angewandte Chemie International Edition in English* 33.23-24 (1995), pp. 2375–2378 (cit. on p. 6).

[18] Facundo Mémoli. "Gromov–Wasserstein Distances and the Metric Approach to Object Matching". In: *Foundations of Computational Mathematics* 11.4 (Aug. 2011), pp. 417–487. URL: https://doi.org/10.1007/s10208-011-9093-5 (cit. on pp. 3, 4, 9–13, 27, 29, 33, 39).

[19] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018 (cit. on pp. 4, 44, 45).

[20] Garth Powis and William R Montfort. "Properties and biological activities of thioredoxins". In: *Annual review of biophysics and biomolecular structure* 30.1 (2001), pp. 421–455 (cit. on p. 6).

[21] Szymon Rusinkiewicz and Marc Levoy. "Efficient variants of the icp algorithm." In: *3dim*. Vol. 1. 2001, pp. 145–152 (cit. on p. 3).

[22] Radu Bogdan Rusu and Steve Cousins. "3D is here: Point Cloud Library (PCL)". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, May 2011.

[23] Robert W. Sumner and Jovan Popović. "Deformation Transfer for Triangle Meshes". In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 399–405. URL: http://doi.acm.org/10.1145/1015706.1015736 (cit. on pp. 3, 9, 27, 30).

[24] C. Villani. *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society, 2003. URL: https://books.google.de/books?id=GqRXYFxe0l0C (cit. on pp. 12, 19).