University of Tübingen
Faculty of Science
Department of Computer Science

# Master Thesis Bioinformatics

## Title of thesis

First Name and Surname

Date

**Reviewers**

Name First Reviewer
(Bioinformatics)
Institute for Bioinformatics and
Medical Informatics
University of Tübingen

Name Second Reviewer
(Field of Study)
Institute of second reviewer
University of Tübingen

**Surname, First Name:**
*Title of thesis*
Master Thesis Bioinformatics
University of Tübingen
Thesis period: dd.mm.yyyy – dd.mm.yyyy

# Abstract

Write here your abstract.

# Acknowledgements

Write here your acknowledgements.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **BLAST** | Basic Local Alignment Search Tool |
| **...** | ... |

# Chapter 1

# Introduction

# Chapter 2

# Material and Methods

## 2.1 Implementation of FracMinHash

The generation of phylogenetic outlines using FracMinHash was implemented using `java-17`, `jloda3` 1.0.0, `splitstree6` 1.0.0 [7] and `openapi-generator-maven-plugin` 6.3.0.

To sketch a sequence, its $k$-mers are read. $k$-mers that include ambiguous bases (that is, bases that are "N" or "n") are skipped. FASTA files can contain multiple sequences, but they are not concatenated and the $k$-mer decomposition is done for each sequence individually, i.e. no $k$-mer spans two or more sequences. The $k$-mers are brought into their canonical form [8, 16] by transorming all characters to uppercase, calculating the reverse complement of the $k$-mer and using the alphabetically smaller value of both.

The $k$-mers are then hashed using a given hash function and a random seed. For this, the `LongHashFunctions` from `zero-allocation-hashing` 0.16 are used. Java `long` values are signed, so the range of the hash function is $[-2^{63}, 2^{63}]$. Thus, the threshold can be calculated using $-2^{63} + \frac{2^{64}}{s}$.

Check: Do I need a full list or just the main contributers? Do I need a reference for each?

Hash values that are lesser than that threshold are kept in the sketch. The sketch is saved in a serialized format on the hard disk. To estimate the Jaccard index, the containment index and from those the distances, the formulas discussed in section **??** were used [6, 8].

The distances are then stored in Nexus file format using the `SplitsTree` 6.0.0-alpha [7] serialization capabilities.

## 2.2 Benchmarking the Hash Functions

To get an understanding of which hash function performs best in terms of runtime, the `jmh` benchmarking framework 1.37 was used to benchmark the following hash functions:

add citation/links to all hash functions

- MurMur3
- XX3
- XX64
- XX128
- City 1.1
- Farm 1.0
- Farm 1.1
- Wy 3
- Metro

While other hash function implementations were considered, the implementation of all hash functions is given by `zero-allocation-hashing` 0.16. The benchmark was executed in throughput mode using a fork count of 2. Besides that, default parameters were used.

## 2.3 Datasets used

To evaluate properties of FracMinHash, analysis was performed using different data sets.

The first dataset (**A**) consists of 128 different *Phytophthora* genomes. The list is taken from [13] without further modifications. Genomes were downloaded from NCBI uding the `datasets` utility [22] using the accession codes listed in that study.

The second dataset (**B**) consists of 72 mtDNA sequences of *Phytophthora* and other Oomycetes of the Peronosporaceae family. The list is taken without further modifications from Supplementary Table 1 from [27]. Genomes were downloaded from NCBI using the Entrez interface [22] using the accession codes listed in that study, appended by the identifier of the most recent version for that accession code.

The third dataset (**C**) consists of all 64 *Phytophthora* reference sequences in the NCBI databasase ("reference") as well as five different query sequences that are typically found in soil samples of avocado orchards that are infected with *Phytophthora cinnamomi* [23]. Those query sequences are divided into two bacterial genomes, two fungal and one *Phytophthora cinnamomi* genome:

- fungal: *Mortierella claussenii* - GCA_022750515.1

- bacterial: *Thermogemmatispora aurantia* - GCA_008974285.1
- fungal: *Venturia carpophila* - GCA_014858625.1
- bacterial: *Pseudomonas syringae* - GCA_018394375.1
- fungal: *Phytophthora cinnamomi* - GCA_001314365.1

The full list of reference sequences can be found in the appendix . The reference sequences were downloaded using the web interface (Filter: reference sequences), the query sequences were downloaded using the `datasets` utility.

ref

## 2.4 Comparison with published Phylogenies

[13] displays a rooted phylogenetic tree in Figure 4 that was generated using the `mashtree` [9, 16]. Unfortunately, the corresponding distance matrix or a serialized version of the tree are not available. Thus, the data needs to be recomputed. For this, the `mashtree_bootstrap.pl` 1.4.6 was applied to dataset A as described by [13], additionally the distance matrix was saved using the `--outmatrix` parameter.

Distance matrices for dataset A using the FracMinHash method were calculated using different combinations of the scaling parameter $s$ and $k$-mer size $k$:

- $k = 19$, $s = 2000$
- $k = 20$, $s = 2000$
- $k = 21$, $s = 2000$
- $k = 25$, $s = 2000$
- $k = 30$, $s = 2000$
- $k = 21$, $s = 500$
- $k = 21$, $s = 1000$
- $k = 21$, $s = 40000$

To ensure comparability with the `mashtree` results, the same hashing function (MurMur) and random seed (42) were used to calculate the sketches [9, 16].

For all distance matrices, SplitsTree 6.0.0-alpha [7] was used to obtain trees, splits and outlines. Trees were calculated using the Neighbor Joining method [21]. Splits were obtained using the Neighbor Net method [3, 4] using the default parameters. Based on this, a phylogenetic outline was calculated [2].

## 2.5 Split difference analysis

To analyse the differences between the phylogenetic outline based on the Mash distances and the outlines based on FracMinHash further, the splits were analysed in details. For this, the splits were exported from SplitsTree 6.0.0-alpha [7] using the plain text format.

citation

Those files were processed with the script `compare_splity.py`. This is based on `python` 3.12 and `pandas` 2.1.4 [15, 24]. All sets of splits are compared pairwise, that is for two sets $\Sigma_A$ and $\Sigma_B$, the following properties are calculated:

- the total sum of weight for all splits in $\Sigma_A$ and $\Sigma_B$, respectively: $\sum_{s\in\Sigma_A}\omega(s)$ and $\sum_{s\in\Sigma_B}\omega(s)$
- the number of splits in $\Sigma_A/\Sigma_B$ and the number of splits in $\Sigma_B/\Sigma_A$
- the total weight difference associated with the remaining splits, i.e. $\sum_{s\in\Sigma_A/\Sigma_B}\omega(s)$ and $\sum_{s\in\Sigma_B/\Sigma_A}\omega(s)$
- the Robinson-Foulds distance [19] of the two sets of splits, i.e. $D_{RF} = \frac{1}{2}|(\Sigma_A/\Sigma_B)\cup(\Sigma_B/\Sigma_A)|$

The script outputs a list of the diverging splits sorted by the weight of the split. Each split is formatted as a list of taxa names on the smaller side of the split sorted alphabetically and joined using the "|" symbol such that the split can be searched in SplitsTree 6 using the regular expression search.

## 2.6 Reproducing phylogenies for shorter sequences

To get an idea of the practical lower boundarys of FracMinHash in terms of input genome size, dataset B was sketched. As this dataset consists of just a single long FASTA file, it was split into its records using `split-fasta` 1.0.0 . The files were then sketched with FracMinHash using $k = 21$, $s \in \{1, 10, 50, 100, 1000\}$ and the MurMur hash function with random seed 42.

check citation

redo with most recent implementation, check sketch sizes. Redo because I don't know with which version the sketches were computed, thus I don't know how to

Given the calculated distances, SplitsTree 6.0.0-alpha [7] was used to obtain trees, splits and outlines. Trees were calculated using the Neighbor Joining method [21]. Splits were obtained using the Neighbor Net method [3, 4] using the default parameters. Based on this, a phylogenetic outline was calculated [2].

## 2.7 Calculating distances of distantly related genomes and genomes with different sizes

To analyse the claim that FracMinHash works better for genomes of different sizes , both reference and query sequences of dataset C were sketched, distances calculated and the results compared.

ensure this claim is cited at least once!

### Sketching using FracMinHash

Reference and query sequences were sketched with $k = 21$, $s = 2000$ using FarmHash with random seeds $rs \in \{10, 20, 30, 40, 50\}$. As the default sketch size for Mash is 10000 [16] and to ensure that it is not the fact that the relevant sketches are just larger, additional sketching with $s = 500$ (which aims to bring the sketch size of the bacterial query sequences to 10000) and $s = 3500$ (which aims to bring the sketch size of the fungal query sequences to 10000) was performed. Using the sketches, the distance matrix was calculated. For this, the implementation was changed such that an additional log is created that lists all empty intersections when calculating the intersection of the two sketches.

### Sketching using MashTree

To obtain a base value to compare agains, `mashtree` 1.4.6 [9, 16] was applied with hash seed $rs \in \{10, 20, 30, 40, 50\}$, `--outmatrix` and the `--save-sketches` parameter. The resulting sketches were then also converted into JSON format using `mash info -d`.

### Visual comparison and split differences

Given all calculated distances, SplitsTree 6.0.0-alpha [7] was used to obtain trees, splits and outlines. Trees were calculated using the Neighbor Joining method [21]. Splits were obtained using the Neighbor Net method [3, 4] using the default parameters. Based on this, a phylogenetic outline was calculated [2].

The splits were then also analysed using the script outlined in Section 2.5.

### Obtaining reference values

As there are no published phylogenies for dataset C that enable a comparison, I need a different ground truth to compare the results of above distance calculation against. As the aim for this experiment is to analyze the influcence

of different genome sizes of distantly related organisms, I have limited this analysis to the five query sequences of dataset C and the reference genome for *Phytophthora infestans* (GCF_000142945.1) as this is one of the largest genomes in the set of reference sequences. For those sequences, I have prepared two different values to compare against.

The first is the Average Nucleotide Identity (ANI) [10] calculated with `OrthoANI` [11] using the default parameters with `BLAST+` 2.14.1 [5] for alignment.

The second is the Average Amino Acid Identity (AAI) calculated with the `Enveomics Collection` online resource [20] using default parameters. As this requires amino acid sequences as input and those are not available for the *Venturia carpophila* and *Phytophthora cinnamomi* sequences, they were predicted using the `gmes_petap.pl` script provided in the `GeneMark ES` 4.72 package [12] using the default parameters and extracted using the `gffread` utility 0.12.7 [17].

### Calculating empty intersections

When the sketch intersections are empty, the estiamted Jaccard similarity is always 0. Mash does not only calculate the sketch intersections, but also intersects that result with the sketch of the union of the two input sketches [16]. This increases the chances of getting a Jaccard estimation of 0.

To test this, I have prepared a the script `get_empty_intersections.py` which takes a list of Mash sketches in JSON format and outputs all pairs for which the numerator of the Jaccard estimation is empty.

The same is possible using the added log output generated by the distance calculation described in Section 2.7.

Using the five different random seeds, we can convert this output into a table that lists the number of empty intersections for each pairwise calculation.

## 2.8 Hash Count analysis

### $k$-mer Coordinates

To analyse the distribution of $k$-mers that are part of a FracMinHash sketch, the coordinates of each $k$-mer were exported when sketching dataset C with $k = 21$, $s = 2000$ and FarmHash with random seed $rs \in \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$. The larger amount of different hash seeds in comparison with the other experiments is to reduce the impact of a single seed further. The coordinates saved for each $k$-mer consist of the following:

- the sequence index, the index of the sequence inside the FASTA file

- the $k$-mer index relative to the file, $k$-mers including ambiguous bases are not counted, no $k$-mer is spanning two or more sequences in the FASTA file
- the $k$-mer index relative to the current sequence, $k$-mers including ambiguous bases are not counted
- the $k$-mer index relative to the file, $k$-mers including ambiguous bases are counted
- the $k$-mer index relative to the current sequence, $k$-mers including ambiguous bases are counted
- the $k$-mer itself, not the canonical form of it
- the hash value of the canonical $k$-mer

The coordinate files were then analyzed with the script `coordinates_complexity_correlation.py` using `python` 3.12 and `numpy` 1.26.3 [1]. For this analysis, each sequence in the FASTA is split into non-overlapping windows of size $w = 10000$. For each window, the number of hashes inside the window that are part of the sketch are counted including duplicates and ambiguous bases.

citation

## Sequence Complexity

Also part of the analysis is the the sequence complexity for each window. For this, the `macle` tool [18] was used. First, the single FASTA file was split into the individual sequences using `split-fasta` 1.0.0 . Then, the complexity was calculated using the default parameters of `macle` and a a window size of $w = 10000$. The output identifies each window by the position of its center base, so this needs to be shifted to the front of the window to match them to the hash count windows. `macle` prints a complexity of $-1$ if the window contains ambiguous bases, thus this information was used to identify and discard those windows.

check citation

## Statistical Analysis of connection between hash counts and sequence complexity

For all windows with non-negative sequence complexity $C_m$, the Pearson-Correlation was calculated.

check format and citation

Next, to check if there is any connection between windows having an unusual hash count and having a low sequence complexity, we need to define what we mean by that: Given the scaling parameter $s$ and a window size $w$,

we would expect on average $\frac{w}{s}$ hashes per window that are part of the sketch. A window has **unusual hash counts** if the number of hashes is outside the interval $[\frac{w}{s} - r\frac{w}{s}, \frac{w}{s} + r\frac{w}{s}]$. I have set $r = 0.95$, $s = 2000$ and $w = 10000$, which classifies all windows with a hash count of 0 or $\geq 10$ as unusual.

Given those inputs, I have performed the Mann-Whitney-U Test [14, 26] with the complexity values of each window into the two categories **unusual count** and **usual count**. I have corrected $\alpha = \frac{0.05}{69}$ to account for multiple testing of the 69 different genomes.

The statistics were computed using `scipy` 1.11.3 [25].

## Analysis of potential connection between low hash counts and gene location

To check if unusual hash counts can be linked to the location of (effector) genes in *Phytophthora infestans*, I have mapped the start position of each CDS annotation to the corresponding window and counted the number of CDS per window.

# Chapter 3

# Results

In this chapter which also could be more than one chapter, depending on the nature of the thesis, the results of the thesis are presented. Make sure you illustrate your results with appropriate figures and tables, but do not discuss the results here. This should be done in a separate discussion chapter.

# Chapter 4

# Discussion

Of course very important! You need to discuss the informatics as well as bio part of your thesis topic.

This section should include the following:

- Short summary of the aim of the study
- Discussion of the results
- Contextualisation of the results/ Putting the results into context (that was set in the introduction)
- You can also discuss limitations of your thesis and formulate an outlook for future/ follow-up research (Outlook can become an extra chapter.)
- Finally, you can add a short conclusion of the main findings/ take aways of the thesis

Take your time for writing the discussion, besides the introduction chapter it is the most important chapter of your thesis.

Also do not subsection the discussion too heavily.

At least 5 pages.

# Appendix A

# Further Tables and Figures

## A.1 List of reference sequences for dataset C

- GCA_000439335.1
- GCA_000443045.1
- GCA_000468175.2
- GCA_000500205.2
- GCA_000500225.2
- GCA_000687305.2
- GCA_001314345.1
- GCA_001314375.1
- GCA_001314425.1
- GCA_002215365.1
- GCA_002812785.1
- GCA_007655245.1
- GCA_008079305.1
- GCA_008080845.1
- GCA_009729435.1
- GCA_011320135.1
- GCA_011947325.1

- GCA_011947335.1
- GCA_011947345.1
- GCA_011947355.1
- GCA_012295415.1
- GCA_012295475.1
- GCA_012656075.1
- GCA_012656105.1
- GCA_014706105.1
- GCA_014706115.1
- GCA_014706125.1
- GCA_014706135.1
- GCA_014706145.1
- GCA_016169955.1
- GCA_016864655.1
- GCA_016880985.1
- GCA_018691715.1
- GCA_018806915.1
- GCA_018873745.1
- GCA_019155715.1
- GCA_020800215.1
- GCA_023611945.1
- GCA_024211575.1
- GCA_024679045.1
- GCA_024679075.1
- GCA_024679115.1
- GCA_024679135.1

- GCA_024679175.1
- GCA_024679225.1
- GCA_024679275.1
- GCA_024679295.1
- GCA_025722995.1
- GCA_030027945.1
- GCA_030267725.1
- GCA_030267785.1
- GCA_030324255.1
- GCA_030463285.1
- GCA_031305395.1
- GCA_032158285.1
- GCA_032432875.1
- GCA_033557915.1
- GCA_033557925.1
- GCA_033557995.1
- GCA_033558005.1
- GCA_033558025.1
- GCF_000142945.1
- GCF_000149755.1
- GCF_000247585.1

# Bibliography


[1] *Array Programming with NumPy | Nature*. URL: https://www.nature.com/articles/s41586-020-2649-2 (visited on 05/13/2024).

[2] Caner Bagci, David Bryant, Banu Cetinkaya, and Daniel H Huson. "Microbial Phylogenetic Context Using Phylogenetic Outlines". In: *Genome Biology and Evolution* 13.9 (Sept. 1, 2021), evab213. DOI: 10.1093/gbe/evab213.

[3] David Bryant and Daniel H. Huson. "NeighborNet: Improved Algorithms and Implementation". In: *Frontiers in Bioinformatics* 3 (Sept. 20, 2023), p. 1178600. DOI: 10.3389/fbinf.2023.1178600. pmid: 37799982.

[4] David Bryant and Vincent Moulton. "Neighbor-Net: An Agglomerative Method for the Construction of Phylogenetic Networks". In: *Molecular Biology and Evolution* 21.2 (Feb. 1, 2004), pp. 255–265. DOI: 10.1093/molbev/msh018.

[5] Christiam Camacho et al. "BLAST+: Architecture and Applications". In: *BMC Bioinformatics* 10.1 (Dec. 15, 2009), p. 421. DOI: 10.1186/1471-2105-10-421.

[6] Mahmudur Rahman Hera, N. Tessa Pierce-Ward, and David Koslicki. *Debiasing FracMinHash and Deriving Confidence Intervals for Mutation Rates across a Wide Range of Evolutionary Distances*. May 30, 2023. DOI: 10.1101/2022.01.11.475870. URL: https://www.biorxiv.org/content/10.1101/2022.01.11.475870v4 (visited on 11/03/2023). preprint.

[7] Daniel H. Huson and David Bryant. "Application of Phylogenetic Networks in Evolutionary Studies". In: *Molecular Biology and Evolution* 23.2 (Feb. 1, 2006), pp. 254–267. DOI: 10.1093/molbev/msj030.

[8] Luiz Irber et al. *Lightweight Compositional Analysis of Metagenomes with FracMinHash and Minimum Metagenome Covers*. preprint. Bioinformatics, Jan. 12, 2022. DOI: 10.1101/2022.01.11.475838.

[9] Lee S. Katz et al. "Mashtree: A Rapid Comparison of Whole Genome Sequence Files". In: *Journal of Open Source Software* 4.44 (Dec. 10, 2019), p. 1762. DOI: 10.21105/joss.01762.

[10] Konstantinos T. Konstantinidis and James M. Tiedje. "Towards a Genome-Based Taxonomy for Prokaryotes". In: *Journal of Bacteriology* 187.18 (Sept. 15, 2005), pp. 6258–6264. DOI: 10.1128/jb.187.18.6258-6264.2005.

[11] Imchang Lee, Yeong Ouk Kim, Sang-Cheol Park, and Jongsik Chun. "OrthoANI: An Improved Algorithm and Software for Calculating Average Nucleotide Identity". In: *International Journal of Systematic and Evolutionary Microbiology* 66.2 (2016), pp. 1100–1103. DOI: 10.1099/ijsem.0.000760.

[12] Alexandre Lomsadze, Vardges Ter-Hovhannisyan, Yury O. Chernoff, and Mark Borodovsky. "Gene Identification in Novel Eukaryotic Genomes by Self-Training Algorithm". In: *Nucleic Acids Research* 33.20 (Nov. 1, 2005), pp. 6494–6506. DOI: 10.1093/nar/gki937.

[13] Kajal Mandal, Subhajeet Dutta, Aditya Upadhyay, Arijit Panda, and Sucheta Tripathy. "Comparative Genome Analysis Across 128 Phytophthora Isolates Reveal Species-Specific Microsatellite Distribution and Localized Evolution of Compartmentalized Genomes". In: *Frontiers in Microbiology* 13 (Mar. 16, 2022), p. 806398. DOI: 10.3389/fmicb.2022.806398. pmid: 35369471.

[14] H. B. Mann and D. R. Whitney. "On a Test of Whether One of Two Random Variables Is Stochastically Larger than the Other". In: *The Annals of Mathematical Statistics* 18.1 (Mar. 1947), pp. 50–60. DOI: 10.1214/aoms/1177730491.

[15] Wes McKinney. "Data Structures for Statistical Computing in Python". In: Python in Science Conference. Austin, Texas, 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[16] Brian D. Ondov et al. "Mash: Fast Genome and Metagenome Distance Estimation Using MinHash". In: *Genome Biology* 17.1 (June 20, 2016), p. 132. DOI: 10.1186/s13059-016-0997-x.

[17] Geo Pertea and Mihaela Pertea. *GFF Utilities: GffRead and GffCompare*. Sept. 9, 2020. DOI: 10.12688/f1000research.23297.2. F1000Research: 9:304. URL: https://f1000research.com/articles/9-304 (visited on 05/12/2024). preprint.

[18] Anton Pirogov, Peter Pfaffelhuber, Angelika Börsch-Haubold, and Bernhard Haubold. "High-Complexity Regions in Mammalian Genomes Are Enriched for Developmental Genes". In: *Bioinformatics* 35.11 (June 1, 2019), pp. 1813–1819. DOI: 10.1093/bioinformatics/bty922.

[19] D. F. Robinson and L. R. Foulds. "Comparison of Phylogenetic Trees". In: *Mathematical Biosciences* 53.1 (Feb. 1, 1981), pp. 131–147. DOI: 10.1016/0025-5564(81)90043-2.

[20] Luis M. Rodriguez-R and Konstantinos T. Konstantinidis. *The Enveomics Collection: A Toolbox for Specialized Analyses of Microbial Genomes and Metagenomes*. e1900v1. PeerJ Inc., Mar. 27, 2016. DOI: 10.7287/peerj.preprints.1900v1.

[21] N Saitou and M Nei. "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees." In: *Molecular Biology and Evolution* 4.4 (July 1, 1987), pp. 406–425. DOI: 10.1093/oxfordjournals.molbev.a040454.

[22] Eric W. Sayers et al. "Database Resources of the National Center for Biotechnology Information". In: *Nucleic Acids Research* 50.D1 (Jan. 7, 2022), pp. D20–D26. DOI: 10.1093/nar/gkab1112. pmid: 34850941.

[23] Itzel A. Solís-García et al. "Phytophthora Root Rot Modifies the Composition of the Avocado Rhizosphere Microbiome and Increases the Abundance of Opportunistic Fungal Pathogens". In: *Frontiers in Microbiology* 11 (2020), p. 574110. DOI: 10.3389/fmicb.2020.574110. pmid: 33510714.

[24] The pandas development team. *Pandas-Dev/Pandas: Pandas*. Version v2.2.2. Zenodo, Apr. 10, 2024. DOI: 10.5281/zenodo.10957263.

[25] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[26] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. DOI: 10.2307/3001968. JSTOR: 3001968.

[27] Richard C. Winkworth et al. "Comparative Analyses of Complete Peronosporaceae (Oomycota) Mitogenome Sequences—Insights into Structural Evolution and Phylogeny". In: *Genome Biology and Evolution* 14.4 (Apr. 10, 2022). Ed. by Liliana Milani, evac049. DOI: 10.1093/gbe/evac049.

# Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum Unterschrift