ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The Structural DNA of Cities

Master Thesis

Franziska Krummenacher

April 8, 2019

Advisors: Monika Niederhuber, Dr. Trivik Verma
Department of Environmental Systems Science, ETH Zürich

Abstract

Since cities are growing faster than ever, city planning is crucial to maintain a fully functional and efficient infrastructure. However, still no comprehensive city model exists that is able to explain the structure of today's cities and predict their future. In order to take a small step towards developing such a model, we aim at identifying the basic building blocks of cities. This thesis proposes a data-driven approach towards city modelling using unsupervised clustering techniques. Complete city maps of 251 cities worldwide are analyzed. First, clustering is conducted on scalar features and a similarity measure between cities. We show that although we obtain reasonable clustering results, this approach is unsuitable for the identification of the fundamental elements of cities. In the second part, we focus on network motifs in city graphs and also use latent Dirichlet allocation, a technique from natural language processing, for in-depth city analysis based on network subgraphs and motifs.

Acknowledgements

I would like to thank my supervisor Dr. Trivik Verma for his guidance and his precious advice. His door was always open if I had any questions which I deeply appreciate.

Furthermore, I would like to express my gratitude to Monika Niederhuber and all other members of the Chair of Land Use Engineering for their constructive and most helpful feedback and inputs.

Finally, this thesis would not have been possible without my partner Marvin, who patiently listened to my thoughts, reminded me of the bigger picture and supported me in any way possible. Thank you very much.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cities are highly complex systems composed of diverse elements such as street networks, power lines, commuting routes and social networks of their inhabitants. They are growing at an unprecedented pace and mostly in an organic, demand-based manner. Furthermore, the lack of global planning leads to inefficiency and susceptibility to widespread infrastructure failure, especially in the event of exceptional occurrences like hurricanes or other natural disasters.

Still, city development is only partially understood and general design plans for sustainable, comfortable and stable cities respecting the limitations of space and resources do not exist. A comprehensive city model that is able to analyze existing cities and predict future city evolution is the first step in order to direct the inevitable changes and growth processes in cities.

This thesis proposes a data-driven approach towards city modelling. More than 90% of human data has been generated in the last two years alone, and with digitization and new technology, data coverage is expected to increase even more. Using techniques from data science and machine learning, we aim at identifying the fundamental elements of cities that determine their characteristics analogously to the base pairs that form human DNA. In the following work, full city maps are analyzed in order to reveal their hidden relations and their core components.

Chapter 2 gives more details about the fundamental ideas behind this work. The data that was used and the data gathering process itself is explained in chapter 3. In chapter 4, features are extracted from the data

based on which clustering is conducted in chapter 5. Then, we present a very different approach based network motifs in chapter 6. The results of both conventional clustering and network motif analysis are finally discussed in chapter 7.

Chapter 2

Motivation and Previous Work

Urban science is currently facing huge challenges. Since cities are growing faster than ever, understanding of infrastructure and city structures is key in order to build efficient cities. Moreover, changing climate conditions and their consequences pose an increasing risk towards cities.[1] Although cities have been studied for a long time, there exists no comprehensive city model [2].

Wide-known features of cities are power laws: Correlations between city population and measures such as geographical features or consumption patterns [2] [3] [4]. Arcaute et al. [5] on the other hand have criticized these power laws and shown that the exponents of the power laws strongly depend on the chosen data. Depersin et al. have also confirmed that scaling laws do not explain the development of cities well [6]. Furthermore, Zipf's law is known to describe the distribution of city sizes [7].

City structure is relatively well-defined and data is easily available, since maps exist for most cities world wide. Although traffic flows are also a core problem of modern cities, comprehensive traffic data is still rare due to the technology for measuring traffic being developed only recently. Furthermore, for detailed traffic analysis data privacy is an important concern while maps are usually publicly accessible, for example via Open Street Maps. Still, a lot of work has already been done on city traffic [8] [9].

In this thesis, a novel approach towards city models is proposed. It aims at identifying the basic building blocks of cities in order to build a comprehensive city model. Instead of finding correlations between

single measures (e.g. population, area etc.), we try to find the elements whose combination defines a city.

Furthermore, the approach presented in this thesis is data-driven. It takes advantage of both the rapidly increasing amount of city data as well as unsupervised clustering methods from the field of machine learning. The aim of this thesis is to explore different city clustering techniques and to find the underlying fundamental structural elements of cities based on the clustering results. Previously, Moosavi has shown that deep learning is able to cluster cities based on satellite images [10].

Analogously to a human body, a city is built from small, fundamental elements that are assembled in a specific frequency and order. We strongly believe that identifying these building blocks allows us to deeply understand and characterize cities.

Chapter 3

Data

In this chapter we present the data that was used in this thesis. The process of choosing suitable cities, downloading city maps and adding population data is described in detail. Furthermore, the structure of the database storing the city maps is outlined and a python toolbox for easy access is briefly presented.

3.1 Data collection process

Data collection was performed in three steps. First, a list of suitable cities was compiled. It contained 350 cities from all continents. Besides the world's largest cities such as Tokyo, New York or Paris, medium and small cities were chosen as well. In total, cities from 76 countries were considered and for each country a minimum of two cities were chosen. For highly populated countries such as the USA or China, up to 15 cities of different sizes were included.

Second, shapefiles and road networks were downloaded for all cities of the list. Shapefiles and networks will be explained in more detail in section 3.2.1. A python download script was written using the package OSMNX [11] which gathers street network data from Open Street Maps. It first searched for a shapefile containing the outline of the city borders and then downloaded all streets lying within these borders. The street network itself was stored as a GraphML-file and as a shapefile. The city outline itself was also stored.

Unfortunately, for some cities, there were no shapefiles for the city borders available. This concerns especially smaller cities in less-developed

areas such as cities in Afghanistan, small towns in India or many cities in African countries. In some cases, there were outlines of the surrounding province instead of city outlines. When the city covered most of the province, this province outline was used instead. About 100 cities however had to be excluded due to the lack of suitable data.

Furthermore, there were some cities where no outline of the complete city was found, but outlines of individual city districts. Since this occurred in a number of large, important cities such as Melbourne, Tokyo and Kinshasa, the complete city outline was built manually by combining all city district outlines, in some cases over 40 districts. Information from governmental sites, encyclopedias and other sources were used to determine which districts to include.

The OSMNX package offers functions to compute statistics based on graph networks, including scalars like the number of nodes and edges as well as histograms of street lengths or node degrees. For all downloaded city street networks, these statistics were computed and stored. More information about these statistics is given in chapter 4.

Moreover, the number of inhabitants was added to the data table. Population data was found on *UNdata* [12]. The most comprehensive city demography data table contains over 55 000 entries for over 4 600 cities all around the world, covering the years 2000-2017. The data distinguishes between female and male inhabitants of a city, as well as population numbers for both city proper and complete urban area for most cities.

From this population data, the total number of inhabitants for each city (using the city proper area if available) was extracted. For each city, the newest figures were considered. In most western cities, these are from 2016 or 2017. However, especially for smaller cities in Africa and Asia, no current data was available and the figures used date back to around 2000. Again, for some smaller cities no UN population data could be found. In these cases, the data table was complemented with numbers from `www.citypopulation.de`. All this, together with the network statistics, resulted in a final data table consisting of 32 columns and 251 rows, where each row represents one city. It was stored as a `.csv` file.

3.2 Data Overview

3.2.1 Shapefiles and GraphML

For each city, its outline was stored as a shapefile. As specified in [13], shapefiles actually consist of at least three files with identical name. The main file with the filename extension `.shp` stores the geospatial data itself as a set of geometric shapes such as lines or polygons. Files ending with `.shx` store the indices of the geometric objects and files with the extension `.dbf` store attributes to the shapes, for example names, scalar data or similar. Many other files can be added to the shapefile in order to extend the contained information.

Not only outlines, but also complete city street networks can be stored as shapefiles. The graph network is first divided into nodes and edges and both of them are then stored in a shapefile of either points or lines. However, this is an inefficient way of storing networks and OSMNX does not directly support loading street networks from shapefiles. Therefore, the graph networks were also stored in GraphML format, which is based on XML syntax. More information about GraphML can be found in [14].

3.2.2 City Distribution

In the final data table, there are 251 cities from 69 different countries. Their geographical location is shown in figure 3.1. The UN categorizes cities by population into four classes which are described in table 4.3. The color of the markers in figure 3.1 indicates the population class of each city. Additionally, figure 3.2 shows the distribution of the population classes in the data set. The colors indicate the population classes, numbered accordingly to table 4.3. We see that African cities are underrepresented, while the density of European cities is very high. Also, there are relatively few cities with very low population count. Both observations can be explained by the fact that maps and population data are available for nearly all cities in western Europe, but are often missing for medium and small cities in Africa, eastern Asia or south America.

3.3 Database

Beside the final data table, a city database consisting of a nested directory structure named *WorldCitiesDatabase* and a set of python functions

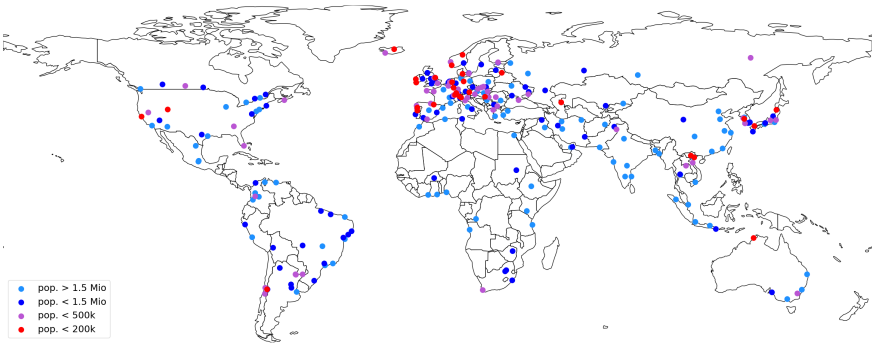

Figure 3.1: Geographical location of all cities in the data set. Colors indicate population classes.

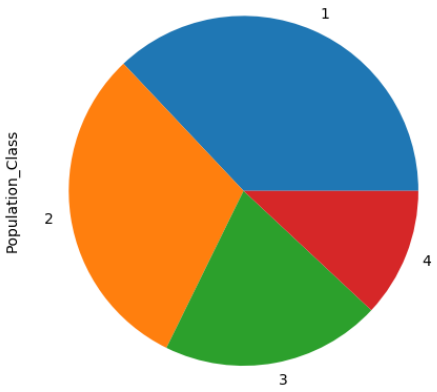

Figure 3.2: Population distribution of cities in data set. The colors represent the population classes, numbered accordingly to table 4.3

.

and tools was set up. Their structure and meaning is described in the following paragraphs.

The layout of the WorldCitiesDatabase is shown in 3.3. For every city, there exists a folder with its name, placed inside the folder of the country to which the city belongs. The country folder itself is again placed in

the folder of the corresponding continent. Each city folder contains five sub folders, which hold the shapefile, the graph network stored in three different ways (GraphML, shapefile and edge list format) and PNG images from the full street network as well as zoomed-in version of the city center. These images are used for visualization of the cities where needed.

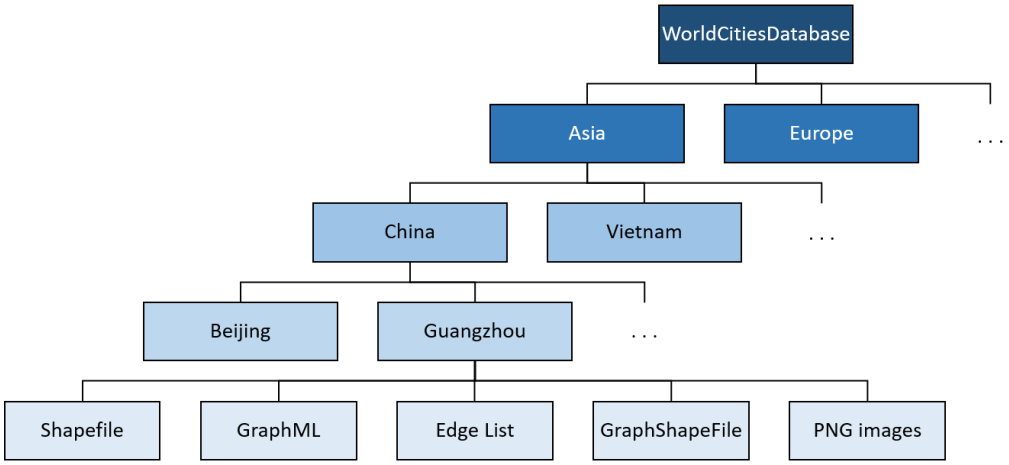

Figure 3.3: Illustration of the city database folder structure. Data for each city is stored in five folders, placed in a folder with the city's name. The city folder itself is stored in a folder for the corresponding country and continent.

The python file `dbtools.py` contains a set of functions that rely on the WorldCitiesDatabase and the data table from section 3.1. They enable simple access to the data files and automate frequently-used procedures such as loading graph files or plotting a set of geographical points to a world map. Most of the python scripts that were written for this thesis build upon these database tools and hopefully they can serve as well for future extensions of this work.

Chapter 4

Feature Identification

This chapter describes three approaches to identify features on the city data explained in chapter 3. Based on these features, the cities will later be clustered by unsupervised machine learning. The first part of this chapter deals with scalar statistics and shows the results of a principal component analysis on these scalar values. The second part considers histogram statistics and shows a method to compare histograms.

4.1 Scalar features

4.1.1 OSMNX statistics

As already mentioned in section 3.1, the OSMNX python package offers two functions that compute statistics based on a given street network. All 251 cities were passed through the functions and the results were stored. For the function `osmnx.stats.extended_stats()`, some of its return values such as eccentricity are computationally very expensive for large networks. Consequently, these were not taken into consideration. Other measures such as pageranks were also excluded since their meaning and relevance could not fully be determined for street networks and they do not comprise basic measures on the street network itself.

From the resulting 25 statistical measures, 18 are scalar values and seven are histograms. The two data shapes were separated and treated individually. In the rest of this section, we confine ourselves to the scalar values. Histograms are treated in section 4.2.

The remaining scalar values cover a wide range of data from the number

variable name	explanation
avg_neighbor_degree_avg	average value of avg_neighbor_degree (see table 4.5)
avg_weighted_neighbor_degree_avg	average value of avg_weighted_neighbor_degree (see table 4.5)
circuity_avg	average circuity
clean_intersection_count	[Always 1, therefore excluded]
clustering_coefficient_avg	average clustering coefficient
clustering_coefficient_weighted_avg	average weighted clustering coefficient
degree_centrality_avg	average degree centrality (see tale 4.5)
edge_length_avg	average edge length
edge_length_total	total edge length
intersection_count	number of intersections
k_avg	average node degree
m	number of edges
n	number of nodes
self_loop_proportion	proportion of streets that form loops
street_length_avg	average street length
street_length_total	total street length
street_segments_count	number of street segments
streets_per_node_avg	average number of streets per node

Table 4.1: Scalar variables returned from OSMNX statistics functions with short explanation.

of nodes and edges in the city graph network to the circuity or the average of histogram data such as the average neighbor degree (i.e. the average number of streets emerging from all neighboring nodes). Table 4.1 shows a list of all statistical variables and their meaning. One of them, *clean_intersection_count*, turned out to be equal to one for all cities and was therefore removed from further analysis.

The scalar population data described in chapter 3 was also included. It encompasses four columns, which are listed and explained in table 4.2.

variable name	explanation
Population_UN	number of inhabitants according to UN population statistics
Year	year of census or population estimate
City_type	area inhabited by the population number from Population_UN. Either *City proper*, *Urban agglomeration* or *Other* (if population data source is other than UN data)
Population_Class	integer from 1 to 4, depending on population size according to table 4.3

Table 4.2: Population data variables with short explanation.

Population Class	Denotation	Population
1	large metro	>1 500 000
2	metro	500 000 - 1 500 000
3	urban	200 000 - 500 000
4	small urban	<200 000

Table 4.3: City classification by population, according to UN Statistics Division. [12]

4.1.2 Correlation

For a first examination of the scalar statistical variables, pairwise correlations are computed. Figure 4.1 shows the correlation values of all combinations. Dark red squares indicate strong positive correlation, dark blue ones denote strong negative correlation.

We clearly see that there are some variable pairs with a very strong correlation. For some of them, such as the number of nodes (n) and the number of edges (m), this is intuitively expected: cities with more streets naturally have more intersections. This is even more true for measures that are very similar in meaning, for example the total edge length and the total street length. Streets with at least one intersection between its start and end point consist of multiple edges, but by definition the total sum of all street lengths is identical to the total lengths of all edges in any city, hence the very strong correlation.

Pairwise correlation can also be visualized by scatter plots where each axis corresponds to one feature and each point represents a city. In figure 4.2, a clear correlation between total street length and population of a city is visible, although there are some outliers. On the opposite, figure 4.3 shows that there is no relationship between average neighbor

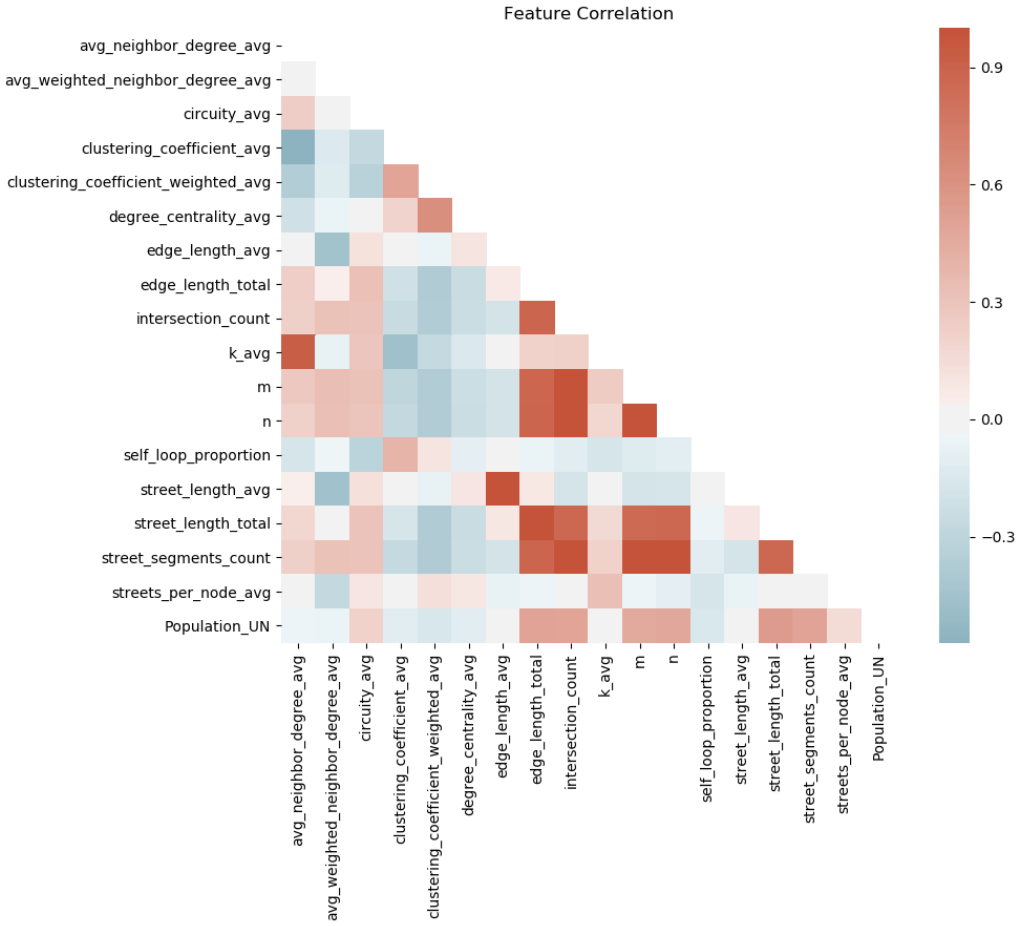

Figure 4.1: Pairwise correlations of all scalar features.Dark red squares indicate strong positive correlation.

degree and average edge length in a city. A very strong correlation can, as expected from the correlation plot, be seen between the total street length and the number of nodes n. This is shown in figure 4.4.

4.1.3 Principal Component Analysis

Subsequently, principal component analysis (PCA) was conducted on the data in order to inspect the importance of the individual features in more detail. PCA is a widely-used technique for feature selection and dimensionality reduction [15] [16].

Consider a feature matrix $\mathbf{X}$ where each column vector represents one data sample in feature space. The process of PCA represents the data

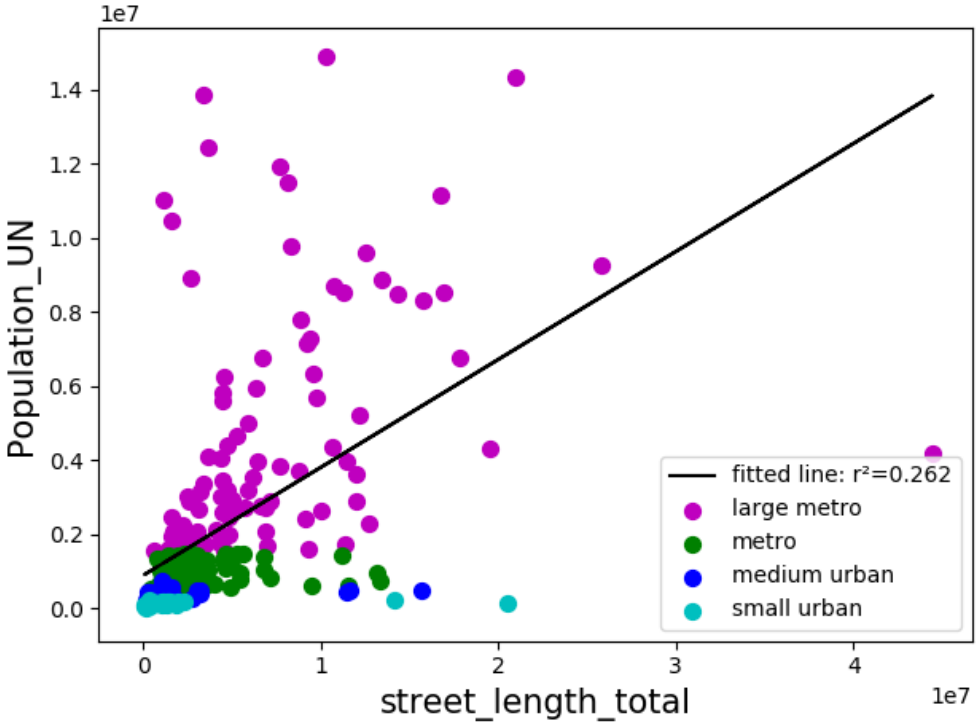

Figure 4.2: Correlation between total street length and population count. Each point represents a city. The colors indicate the population class. The correlation is indicated in the legend.

points with respect to so-called *principal components*. Unlike the features, the principal components are required to be uncorrelated (i.e. orthogonal). It can be shown that this is fulfilled by using the eigenvectors of the covariance matrix $\mathbf{S}$ of the feature matrix $\mathbf{X}$ as a basis for the PCA space [17] [15], for example via singular value decomposition (SVD) [16]. The principal components are then linear combinations of the original features and their total number is identical to the number of features.

In order to perform dimensionality reduction, the principal components are ordered by their corresponding eigenvalue of $\mathbf{S}$. [17] shows that the component corresponding to the largest eigenvalue covers the largest fraction of total data variance. By choosing the n principal components with the largest eigenvalues and neglecting the others, one can reduce the data dimensionality to n dimensions while preserving as much data variance as possible.

Figure 4.5 shows the percentage of variance explained by each principal

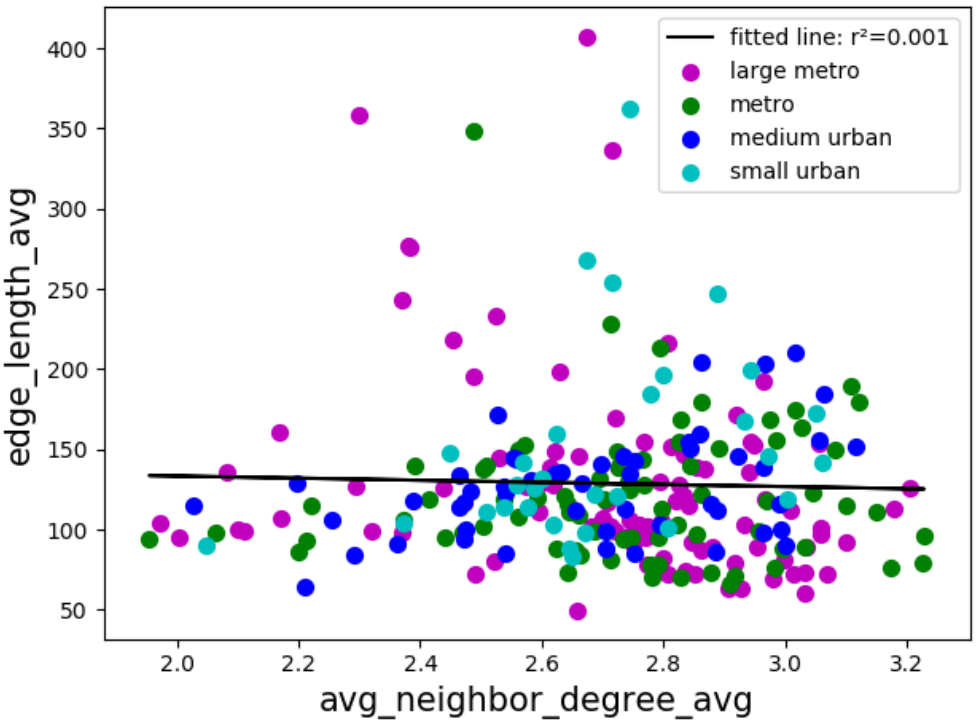

Figure 4.3: Correlation between average neighbor degree and average edge length. Each point represents one city. The colors indicate the population class. The correlation is indicated in the legend.

component found in the city data. The blue line in the plot shows the cumulative variance. We see that if we take the first four components, they together cover about 85% of variance in the data, which is often considered to be enough for practical purposes. Therefore we can reduce the dimensionality of the data to four dimensions.

Since principal components are linear combinations of feature values, PCA value c_i for city i is computed in the following way:

$$ci = \sum_{n=1}^{N} w_n x_n^i, \tag{4.1}$$

where N denotes the number of features and x_n^i is the feature value of city i and feature n. w_n^i is the weight of the n-th feature for the i-th principal component.

In table 4.4 the values of the weights for the first four principal components are listed. We see that all features contribute considerably to at

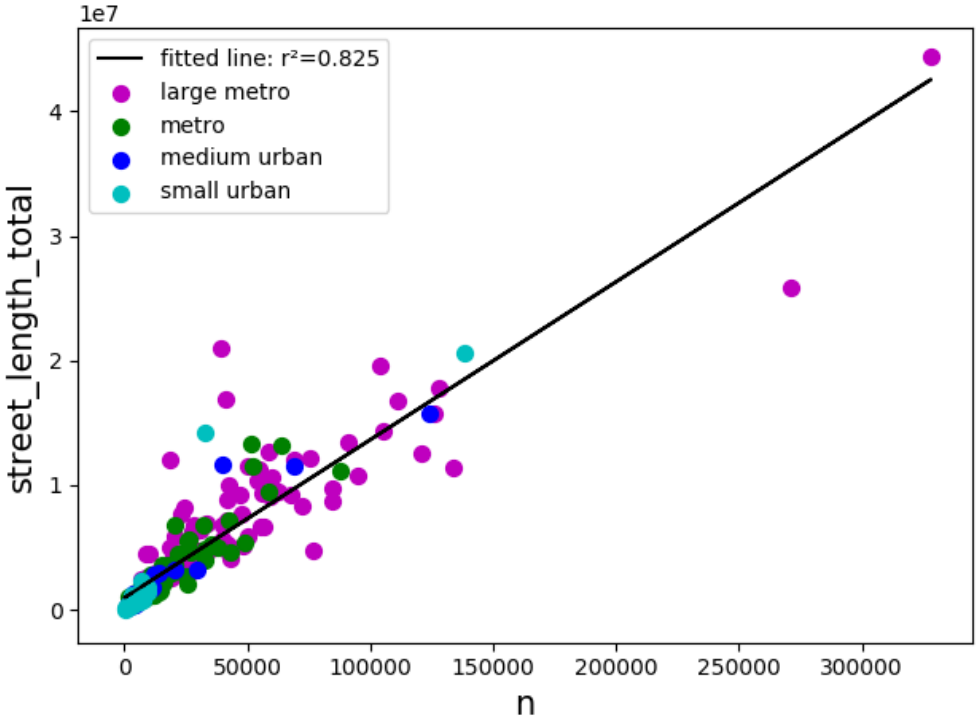

Figure 4.4: Correlation between total street length and number of nodes. Each point represents one city. The colors indicate the population class. The correlation is indicated in the legend.

least one component.

It is important to note that *population_class* and *population_UN* were not taken into account for PCA. If population is considered, it adds a lot of variance to the data and the first principal component consists of only population (i.e. all other weights are close to zero). While this is interesting to observe, it does contradict the aim of PCA and therefore, population was excluded.

Figure 4.6 shows a 3D scatter plot of all cities in PCA space. The three dimensions are the first three principal components of the data set. Each dot represents a city and its color indicates the population (which is, however, not included in PCA). We see that although we did not include population into PCA, cities with similar population sizes seem to cluster to each other. However, the clusters overlap strongly and are widely spread out. For comparison, figure 4.7 shows the distribution of cities in the first two principal components with population data included. It

Feature	1st comp.	2nd comp.	3rd comp.	4th comp.
avg neighbor degree avg	-0.1258	0.1693	0.4566	0.2834
avg weighted neighbor degree avg	-0.0839	-0.4888	-0.0487	0.1314
circuity avg	-0.1735	0.0119	0.1849	-0.1131
clustering coefficient avg	0.1632	0.1255	-0.4081	-0.2558
clustering coefficient weighted avg	0.2537	0.0874	-0.0739	-0.2790
degree centrality avg	0.3425	0.0250	0.1640	0.0215
edge length avg	0.0824	0.5287	-0.1114	0.1987
edge length total	-0.3329	0.1782	-0.1046	0.0299
intersection count	-0.3485	0.0076	-0.1041	-0.0967
k avg	-0.1185	0.2170	0.4743	-0.0510
m	-0.3520	0.0164	-0.0694	-0.0302
n	-0.3489	-0.0039	-0.1163	-0.0261
self loop proportion	0.0596	0.0655	-0.4551	0.0922
street length avg	0.0712	0.5362	-0.0943	0.1989
street length total	-0.3308	0.1740	-0.1327	-0.0254
street segments count	-0.3490	0.0090	-0.1026	-0.0857
streets per node avg	-0.0105	0.1657	0.1855	-0.7974

Table 4.4: Weights for the first four principal components of the scalar features, rounded to four decimals.

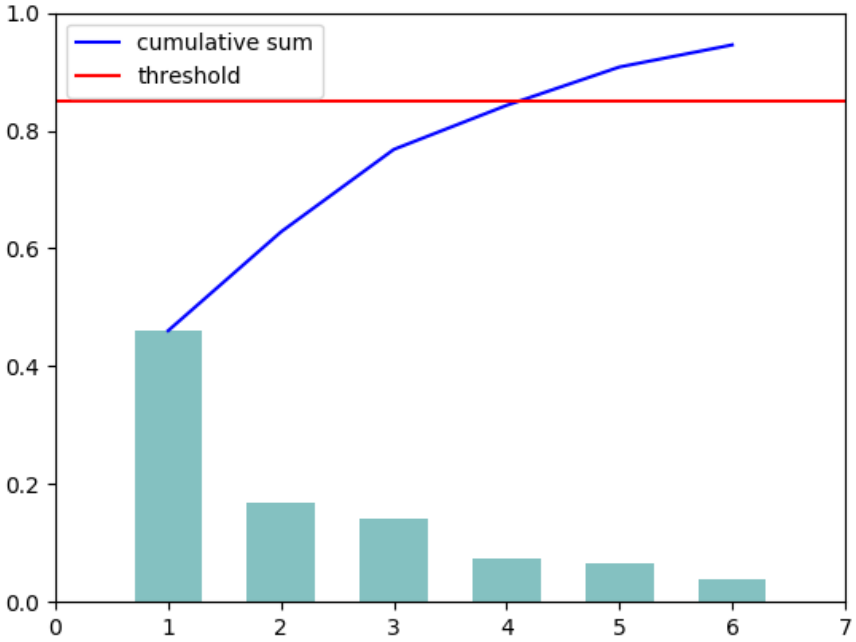

Figure 4.5: Screeplot of pca analysis. The x-axis numbers the components, the y-axis denotes the amount of variance explained by each component. The red line marks the threshold of 85% explained variance. The line in blue is the cumulative sum of the components.

is obvious there that the data is ordered by population.

4.2 Statistical Analysis

4.2.1 OSMNX Histogram features

Besides the scalar features discussed above, the statistics functions of the OSMNX package also return histogram valued features. In contrast to scalar features, histogram features consist of a vector of real values forming a histogram for each single city. Usually, they represent the distribution of a scalar feature in a single street network, where the bins of the histogram correspond to manifestations of this feature and the values are the number of occurrences. The histograms returned by OSMNX are stored in python dictionaries.

Table 4.5 shows a list of all seven histogram-valued features and a short

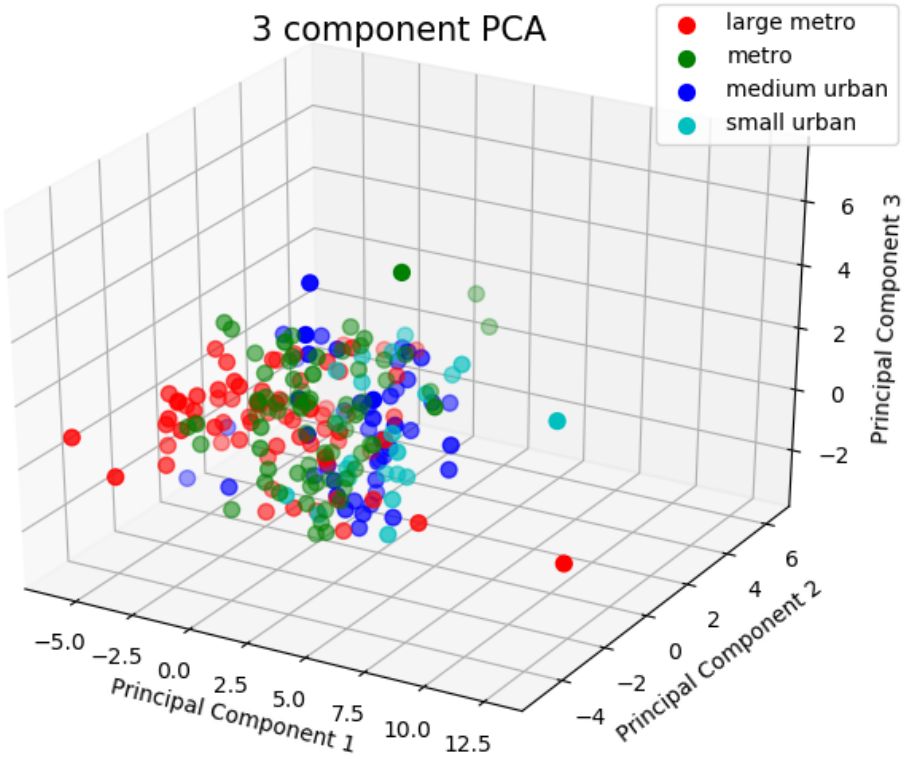

Figure 4.6: Scatter plot of the data points in PCA space (first three dimensions).

description of their meaning. It is important to note that the histograms are not normalized (except for *streets_per_node_proportion*) and each bin represents a single number instead of covering a range. Furthermore, two histograms that cover the same feature but originate from different cities may be completely disjoint. Imagine a perfectly grid-based city where each node is connected to exactly four other nodes. If we neglect the city border nodes, its *avg_neighbor_degree* histogram would consist of a single bar at $x = 4$ and height equal to the number of nodes in the city. Another hypothetical city, might only contain edges with an average neighbor degree of exactly three or five. The two histograms of these two cities would neither have the same number of bins nor share any bin.

In chapter 5, scalar and histogram data are clustered. Simply put, clustering of data points is conducted by grouping close or similar points together. For data points with a given number of scalar features, this is straight-forward: The data points lie in a multidimensional space, each dimension corresponding to one feature. Similar points (i.e. points with

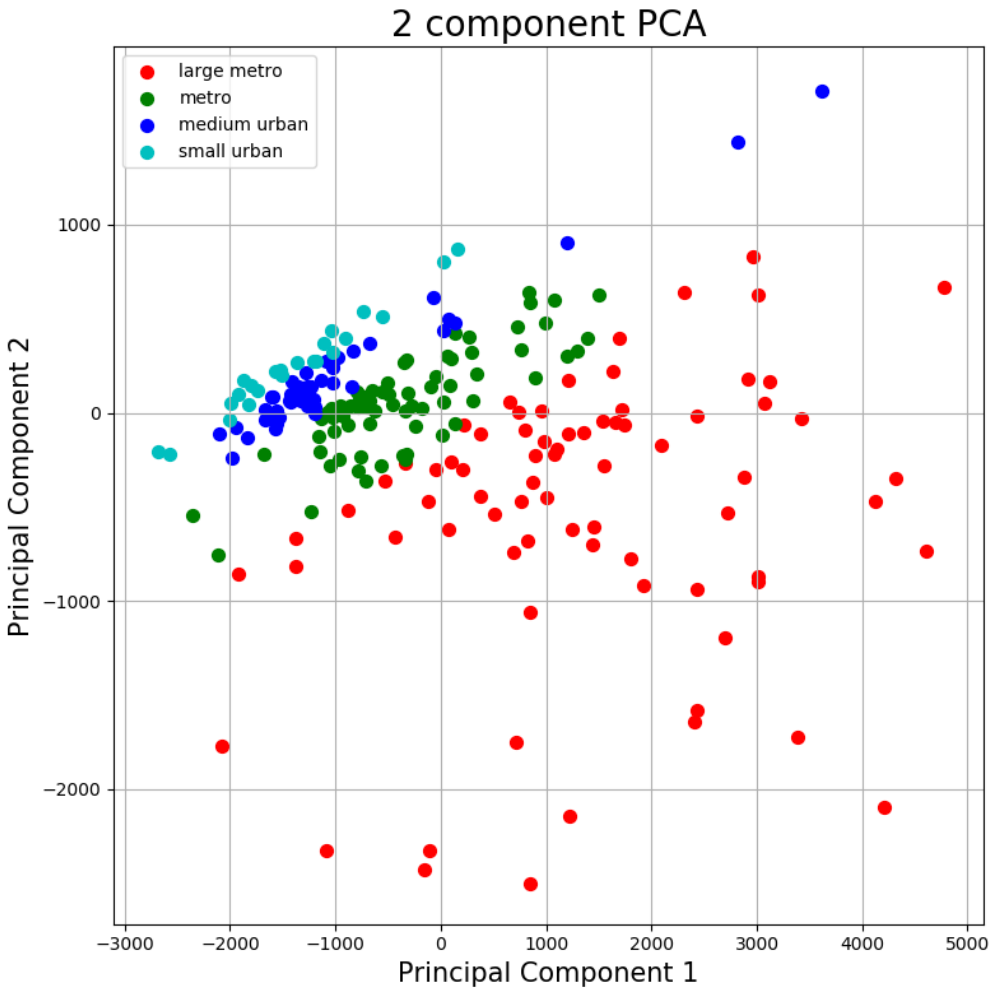

Figure 4.7: Scatter plot of the data points in PCA space (first two dimensions) with population data included.

similar coordinates) can easily be identified by computing for example euclidean distances.

However, this task is much more challenging if the features are not scalars but histograms. There exists a variety of approaches and similarity metrics for histograms, the one used in this thesis being presented in the next section.

histogram feature name	explanation
avg_neighbor_degree	key = average degree of neighbor nodes, value = counts
avg_weighted_neighbor_degree	key = average weighted degree of neighbors, value = counts
clustering_coefficient_weighted	key = weighted local clustering coefficient, value = counts
clustering_coefficient	key = local clustering coefficient, value = counts
degree_centrality	key = degree centrality values value = counts
streets_per_node_counts	key = numbers of streets (edges in undirected graph) emerging from a node, value = counts
streets_per_node_proportion	key = numbers of streets (edges in undirected graph) emerging from a node, value = relative counts

Table 4.5: Histogram variables returned as dictionaries from OSMNX statistics functions with short explanation.

4.2.2 Wasserstein Distance

In this thesis, we measure the similarity between two histograms by a measure called *Wasserstein distance*. The reason for this choice is that in street networks, all histogram bins contain valuable information and Wasserstein distance does include the full distributions. Since the most general formulations of the Wasserstein distance are very abstract [18], we will limit ourselves to a simpler version, which is sufficient for our needs. In particular, the histograms returned by OSMNX are well-defined, have a finite number of bins and finite values at all bins.

Given a set of probabilities $\Gamma(u, v)$ on $\mathbb{R} \times \mathbb{R}$ with marginals u and v, the Wasserstein distance d is defined in [19] [20] as

$$d = \inf_{\pi \in \Gamma(u,v)} \int_{\mathbb{R}\times\mathbb{R}} |x - y| \pi(x, y). \tag{4.2}$$

In our case, u and v are simply the histogram values.

More intuitively, the Wasserstein distance can be understood as follows: Imagine the two histograms being piles of earth. The Wasserstein distance measures the amount of work needed for transforming the first histogram into the second by moving earth from one bin to another. Based on this analogy, the Wasserstein distance is also called *earth-movers distance*.

A python script was written that computes the Wasserstein distance for all histogram features. It uses the function *scipy.stats.wasserstein_distance()* from the scipy package. For each feature, a 251×251 distance matrix A was computed. The entry a_{ij} is equal to the Wasserstein distance between the histograms of city i and j of a feature. The resulting matrix was stored in a file and later used for clustering as described in chapter 5. In total, there were seven distance matrices.

Chapter 5

Clustering

The features that were identified and explained in the last chapter were later used for clustering. The aim of clustering techniques is to group data points based on their position in feature space and identify their underlying structures and patterns. The following two sections will first cover simple clustering on the scalar features and then introduce a more enhanced clustering approach based on similarity graphs which can also include histogram valued features.

5.1 Scalar Clustering

In order to perform unsupervised clustering on our data points in PCA space, two different algorithms were used: K-means and DBSCAN. They are briefly presented in the following sections. Afterwards, the results of both algorithms are shown and discussed.

5.1.1 Clustering Algorithms

K-means

The K-means algorithm is a widely used non-probabilistic clustering method. It is based on the idea that a cluster is a set of points such that the distance between two points belonging to the same cluster is smaller than the distance between two points from different clusters. More formally, this corresponds to a minimization of the cluster variance.

Consider a set of N data points $X = x_1, ..., x_N$ and a given number of clusters K. The clusters can be represented by their mean points μ_k for

$k = 1, ..., K$. Each data point is assigned to a cluster via a binary variable r_{nk}, which is either 0 or 1. If $r_{nk} = 1$, then x_n belongs to the cluster with mean μ_k and since every point can only be assigned to a single cluster, it must hold that $r_{nj} = 0$ for $j \neq k$. Using this notation, K-means clustering is equivalent to the minimization of the function J given as the sum of squared distances of each data point to its assigned cluster center μ_k:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{5.1}$$

J is minimized in a iterative manner. After initializing the μ_k (often as randomly chosen points from X), all data points x_n are assigned to their closest cluster center. Afterwards, the cluster centers are updated by computing the mean point of all points belonging to a given cluster. These two steps are repeated until the cluster assignments converge.

One of the major issues about the K-means algorithm is the input parameter K, i.e. the number of clusters. K needs to be known prior to clustering, which is not always the case or difficult to obtain. If the number of clusters is not known, it makes sense to run K-means for different K and compare the results. This was done for this thesis.

DBSCAN

The name of the second algorithm, DBSCAN, stands for *Density Based Spatial Clustering of Applications with Noise* and was first introduced by Ester et al. [21]. Cluster identification in DBSCAN relies on density measures. It uses two parameters, ε and M, while in practice, M is often set as $M = 4$ [21].

The *ε-neighborhood* $N_\varepsilon(p)$ of a data point p is defined by the set of all data points whose distance to p is smaller than ε. Furthermore, a point p is directly density-reachable from a point q if it is in the ε-neighborhood of q and there are at least M points in this neighborhood.

Roughly speaking, the algorithm works as follows: DBSCAN starts with a random point x in the data set and finds all points p that are either directly density-reachable from x or for which there exists a chain of points $p_1, ...p_n$ starting from x and ending in p such that every p_{i+1} is directly density-reachable from p_i. All of these points are assigned to the same cluster and a new point from the remaining set of unassigned points is drawn. This is repeated until all points have been classified or visited at least once.

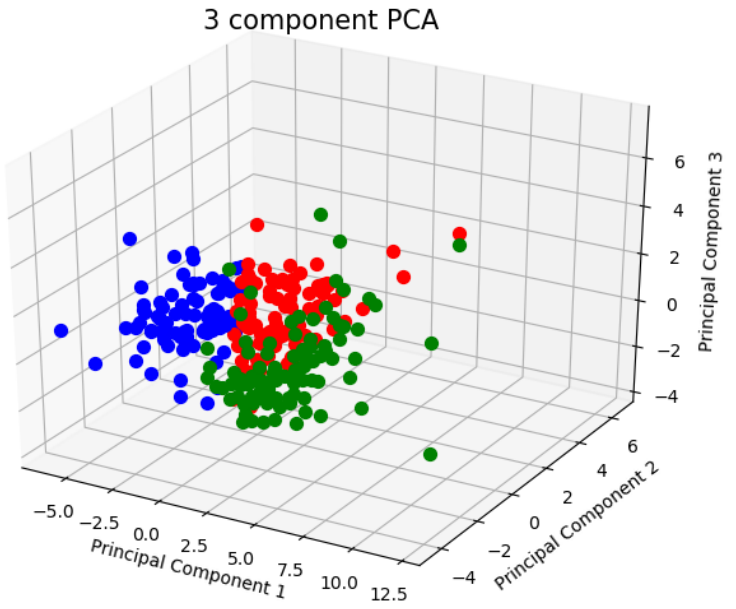

Figure 5.1: Result of K-means clustering with $K = 3$.

The main advantage of DBSCAN is that the number of clusters is detected automatically and no prior knowledge is required. However, close clusters might merge together if the distance between their borders is smaller than ε. To avoid this, it is usually run recursively while ε is decreased in each run.

5.1.2 Results

Results of K-means

In this thesis, the implementation in the python package *sklearn* was used for K-means clustering. Since the number of clusters was not known, it was run for $K = 2, ...6$ on the data in four-dimensional PCA space from section 4.1.3. Figures 5.1 and 5.2 show the resulting clusters for $K = 3$ and $K = 4$. We see that for three clusters, their distribution is similar to the population distribution in figure 4.6, except that population class 3 and 4 are merged into one cluster. If we set K to 4, however, figure 5.2 implies that the detected clusters are about quarters of the data point cloud and do no longer resemble the population classes.

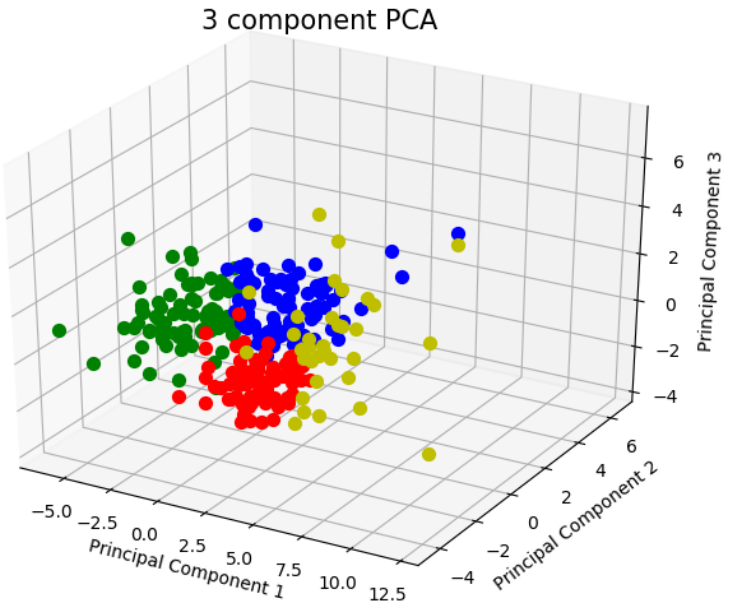

Figure 5.2: Result of K-means clustering with $K = 4$.

Results of DBSCAN

Again, the python package *sklearn* was used in order to apply DBSCAN to the PCA data. As shown in 5.3, the algorithm identifies only a single cluster which contains all data points. This tells us that the data points are roughly homogeneously distributed in this cluster and, as visual inspection also tells, there are no sharply separated regions or clusters.

5.2 Histogram Clustering

5.3 Similarity Graphs

As explained in section 4.2, clustering based on histogram-valued features is not straight-forward. A naïve approach is to treat each bin of each histogram as one dimension in feature space. However, this can result in an extremely high-dimensional clustering problem which is very difficult to solve and prone to overfitting, as euclidean spaces distort in high dimensions.

Furthermore, as also mentioned in section 4.2, the histograms returned by the statistics functions of the osmnx package may be disjoint or only a small fractions of their bins might be non-zero in both histograms.

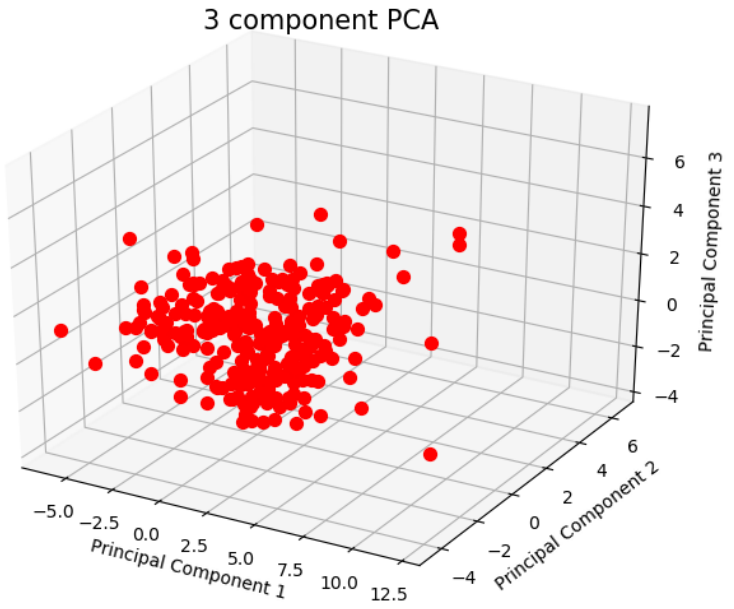

Figure 5.3: Result of DBSCAN clustering.

Having high-dimensional data points with lots of coordinates being zero is clearly not advantageous for clustering.

To avoid this, similarity graphs were constructed from the distance matrices computed in section 4.2. A similarity graph is a fully connected, weighted graph where each node corresponds to a city in the data set and the weights of the edges correspond to the similarity of the two cities connected by the corresponding edge. In total, there were seven similarity graphs for the seven histogram features. To obtain similarity instead of distance, the Wasserstein distance was first normalized: $\hat{d}_{ij} = \frac{d_{ij}}{\max d_{ij}}$. Similarity s_{ij} between two cities i and j was then computed simply as $s_{ij} = 1 - \hat{d}_{ij}$.

We will see in section 5.3.1 how clustering on such similarity graphs can be conducted. However, since each of our distances originates from a single feature, we would receive seven independent clustering results based on one feature each, which has only limited significance. Instead, we collapse the seven similarity graphs into one and cluster on the union of all features.

The construction of this final similarity graph is illustrated in figure 5.4. Given two cities i and j, the weight w_{ij}^n of the edge connecting them is taken from each graph. The superscript n denotes the feature graph

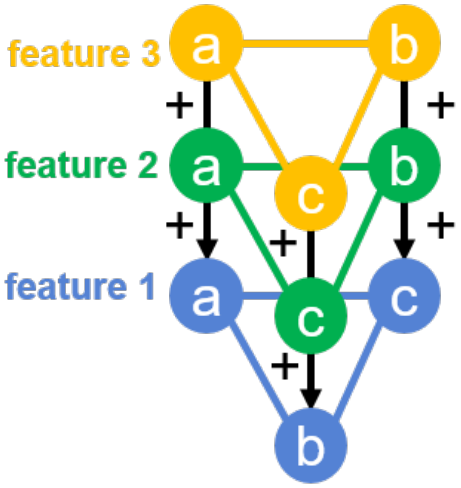

Figure 5.4: Illustration of similarity graph collapsing process.

to which the weight belongs. From these edge weights, the total edge weights are computed as follows:

$$w_{ij}^{tot} = \sqrt{\sum_{n=1}^{7} (w_{ij}^{n})^2} \tag{5.2}$$

Many other ways of computing the total similarity are also imaginable. Equation 5.2 was chosen in analogy to the definition of the absolute value of vectors. If we think of the features as dimensions and of the edge weights as components of a similarity vector, the total similarity value could be defined as the absolute value of this similarity vector.

After collapsing all histogram features into a single similarity graph, the question arises whether it is possible to include also the scalar features and to obtain a similarity graph which contains all available features. For each scalar feature, we can compute a distance matrix by simply storing the feature value difference of all pairs of cities. Then we proceed analogously as for the histogram features, compute the total weights of all histogram and scalar features and obtain a similarity graph which comprises 25 features in total. Please note that again population counts and class labels were not used as features in this step.

5.3.1 Network Clustering Algorithms

After we have constructed our total similarity graph, we can start clustering. Two different algorithms were used for this task: Louvain community detection and spectral clustering. Similar to clustering of the scalar features in PCA space, one of the methods (Louvain community detection) determines the number of clusters itself, while for the other one (spectral clustering), the number of clusters needs to be known in advance.

Louvain Community Detection

Louvain community detection is a method for partitioning a given network into smaller sub-networks that was first presented by Blondel et al. in [22]. A network community is defined as a set of densily connected nodes with only sparse connections to nodes outside the community. Finding the optimal community partition of a given network is computationally hard and therefore a variety of approximate techniques for this task is known.

The Louvain method is based on partition modularity $Q \in -1, 1$ which compares the density of connections inside communities to the density of edges between communities. Formally, modularity is defined as

$$Q = \frac{1}{2m} \sum_{i,j} [w_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j). \tag{5.3}$$

Here, w_{ij} denotes the edge weight between nodes i and j, m is the sum of all edge weights, k_i denotes the sum of the weights of all edges that belong to node i and c_i is the community to which node i belongs. [22]

First part of the algorithm is the initialization, where every node of the network is assigned to a different community. Then, we loop over all nodes and assign them to the community of a neighboring node such that the modularity increases the most. If all possible changes in assignment to another community would result in decreasing modularity, the node remains in its original community. This is repeated for all nodes multiple times until no further improvement in modularity is possible.

Afterwards, a new network is constructed where each node corresponds to a community that has been found in the step before. Edge weights are computed by summing the weights of all edges between two communities. On this new network, one could again apply the first part of

the algorithm and iteratively obtain a hierarchical community structure of the original network.

Spectral Clustering

As in Louvain community detection, spectral clustering aims at determining clusters of nodes in the graph such that the internal edges of each cluster have high weights and the edges between clusters have low weights. The basic principle behind spectral clustering on similarity graphs is to compute the Laplacian matrix L:

$$L = D - W \tag{5.4}$$

Hereby, W denotes the weighted adjacency matrix of the similarity graph and D is a diagonal matrix with d_{ii} being the sum of the weights of all edges touching node i. There exist two normalized forms of L, named L_{sym} (since it is symmetric) and L_{rw} (which is connected to random walks).

The spectral clustering algorithm takes the number of clusters k as input. It then computes the first k eigenvectors of L or one of its normalized forms. The eigenvectors are used as columns to construct a matrix U, whose row vectors are then clustered via k-means. The detailed proofs and explanations of the steps of spectral clustering go beyond the scope of this thesis, but can be found in [23] which does not only give a very profound insight into spectral clustering but also provides further information about different similarity graphs.

5.3.2 Results

Louvain Clustering Results

Despite our greatest efforts, it returned either one single cluster containing all cities or, with changed parameters, there were hundred or more clusters each containing between one and three cities. Both results are useless, since we cannot derive any city characteristics from such a partition.

After numerous attempts we managed to tune the parameters of the function in such a way that it produced a sensible number of clusters. However, the parameters needed to be set very exactly and a discrepancy of just 0.1% of the parameter value already put us back to one or hundreds of clusters. It seems extremely doubtful whether such clustering results can be used for any purpose since such fine tuning implies

that we overfit the model drastically to our expectations. Therefore, we also refrain from showing detailed results from Louvain clustering.

Spectral Clustering Results

Spectral clustering was conducted by using the corresponding `sklearn` function. As already mentioned, the number of resulting clusters k needs to be given. Since this was not known in advance for our city data set, all possibilities between two and six clusters were tried.

Contrary to the Louvain community detection algorithm, spectral clustering returned fairly balanced clusters with similar numbers of cities. In order to visually inspect the results, six cities were drawn at random from every cluster and their map as well as zoomed-in images of the city centers were displayed as a group.

Figures 5.5, 5.6 and 5.7 each show the complete maps of six sample cities for the case $k = 3$. Additionally, figures 5.8, 5.9 and 5.10 show extracts of the city centers for the three clusters. For reasons of clarity, the results for $k = 2, 4, 5, 6$ are not shown here, but can be found in appendix A.

We can see some tendencies in these clusters: The first cluster seems to contain mainly strongly grid-based cities and no megacities. The former is even more visible in the enlarged city centers (figure 5.8). The second cluster contains tendentially large cities such as Shanghai or Rio de Janeiro. We clearly see the grid-structure in their city centers, but they appear less regular than in the smaller cities from cluster one. Looking at the third cluster, it is obvious from figure 5.7 that these are smaller cities than the ones in the second cluster and they are all European cities. The enlarged maps show a much more diverse, probably historically grown city center structure.

All in all, these results look promising and show that different classes of cities can be identified using the complete set of scalar and histogram data. However, we have not come closer to our goal which is to determine the underlying building blocks of cities. The results of our clustering heavily depend on the set of features. If we removed the scalar features or added ten new measures, the clusters would change fundamentally. Furthermore, the features are not weighted in our similarity graph, meaning that all of them are equally important. It is hence impossible to distinguish between important and unimportant features. Therefore, we will try a very different approach based on the network

structure itself in the next chapter. A more detailed discussion of the clustering results can be found in section 7.2.

Figure 5.5: Spectral clustering results: Random samples from first cluster of three.

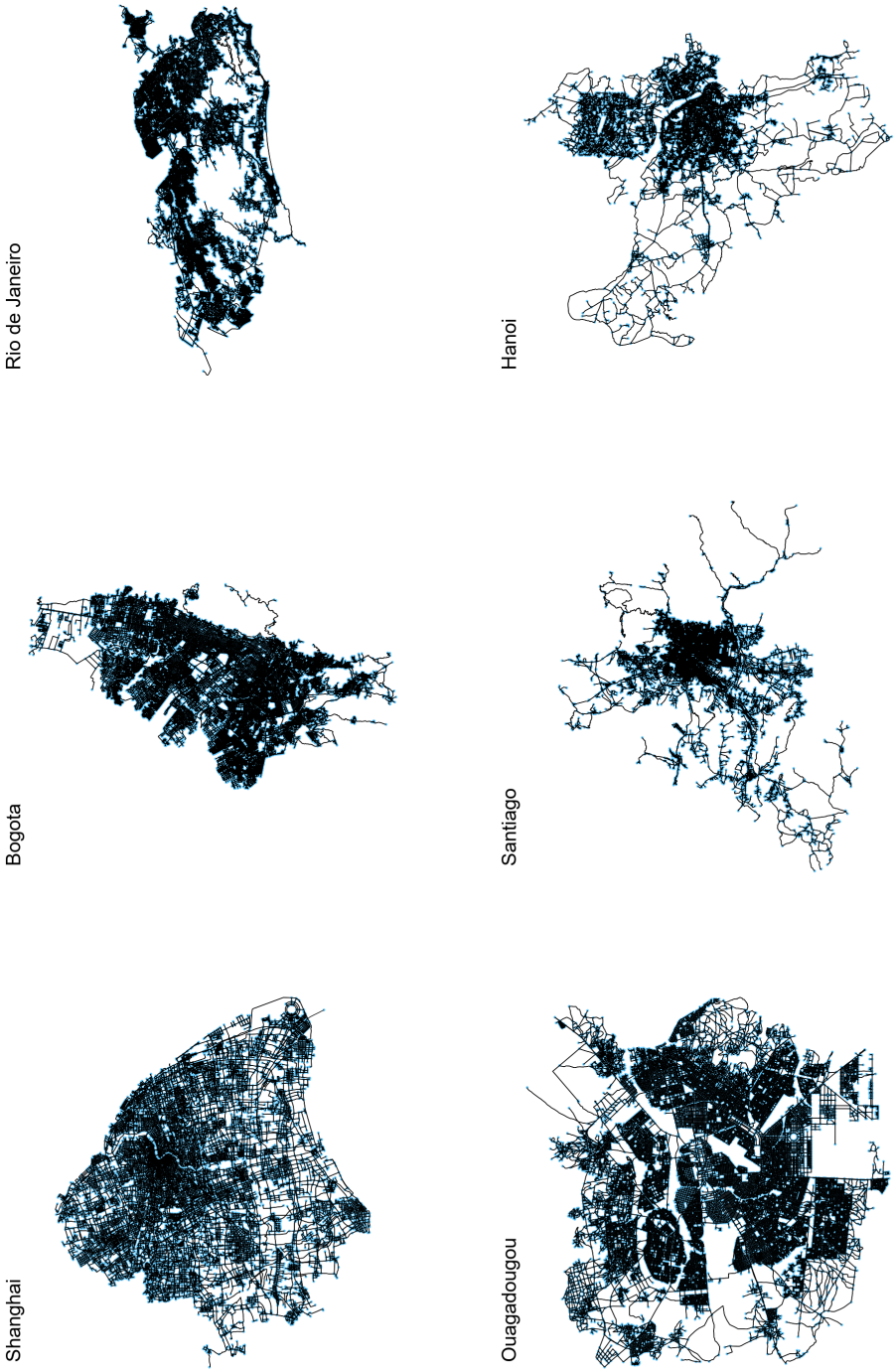

Figure 5.6: Spectral clustering results: Random samples from second cluster of three.

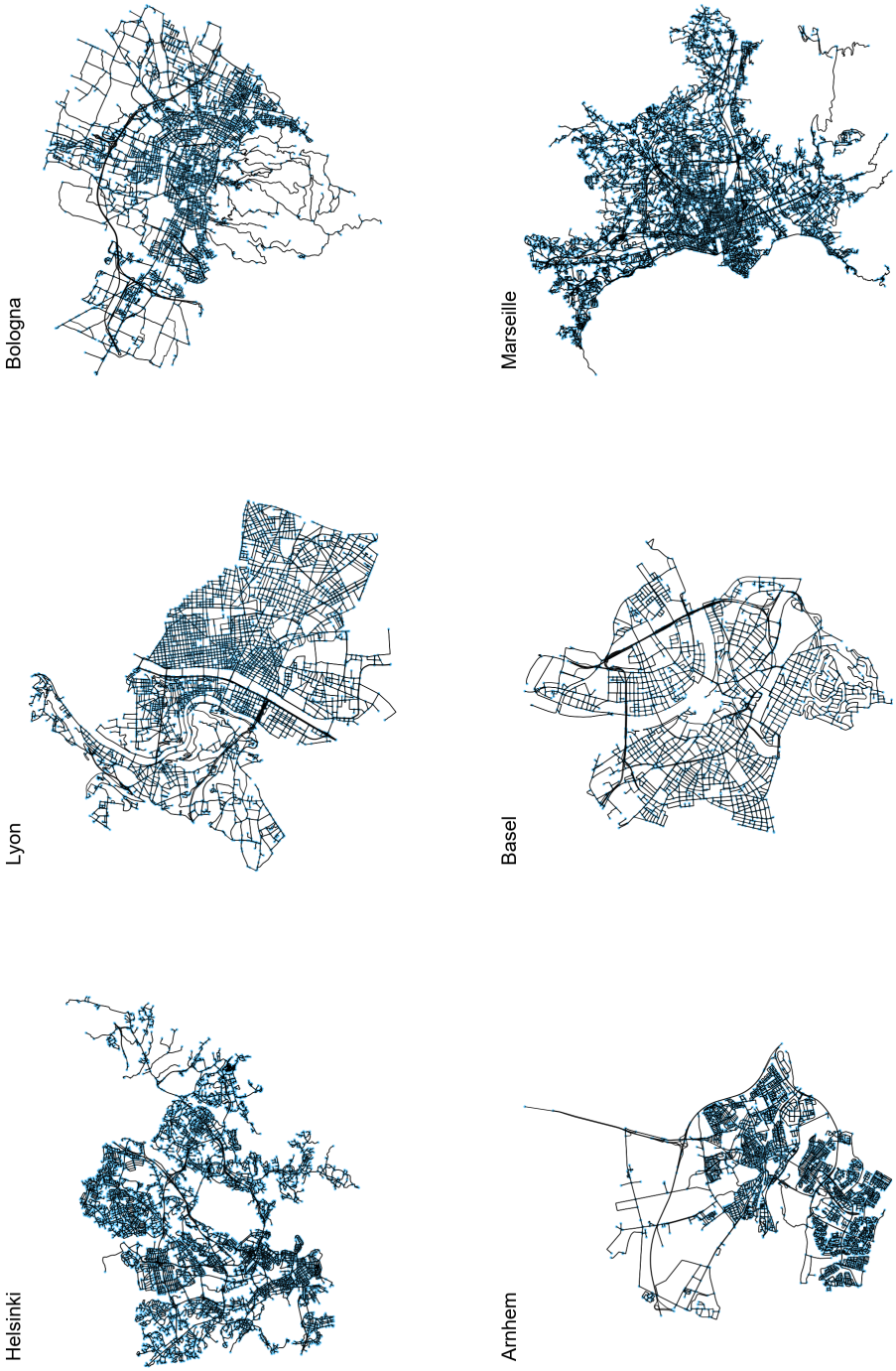

Figure 5.7: Spectral clustering results: Random samples from third cluster of three.

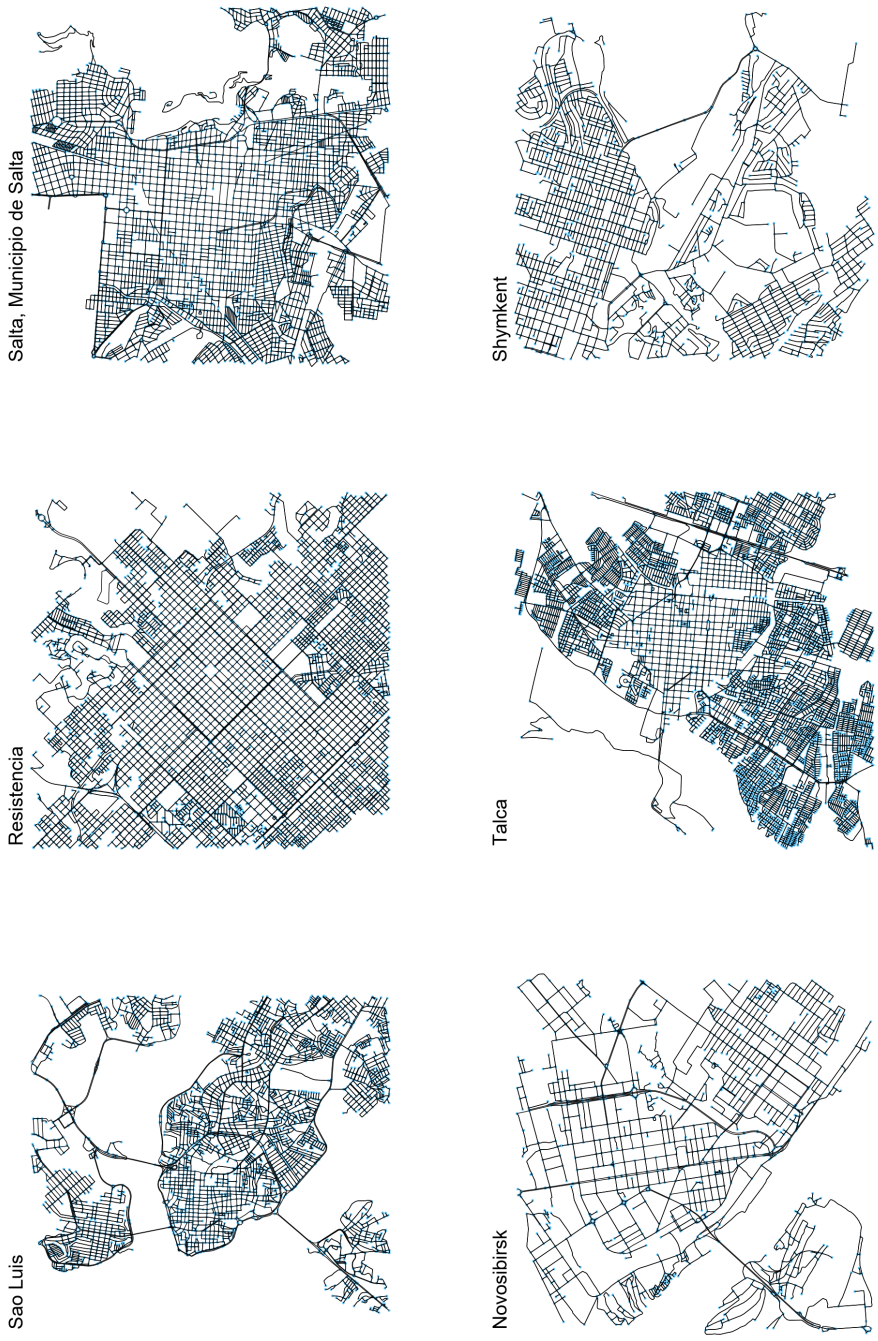

Figure 5.8: Spectral clustering results: Random samples from first cluster of three (enlarged city centers).

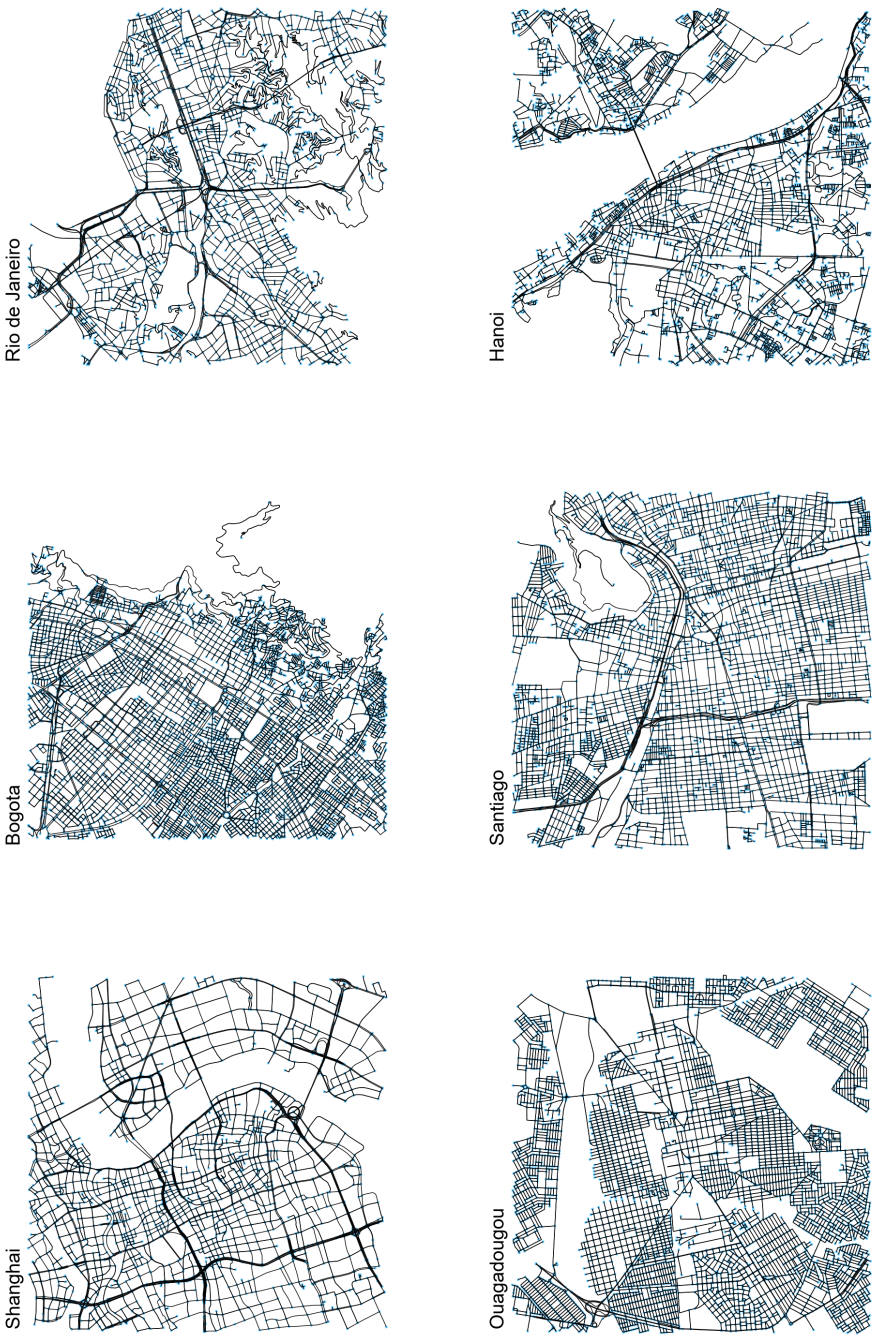

Figure 5.9: Spectral clustering results: Random samples from second cluster of three (enlarged city centers).

Figure 5.10: Spectral clustering results: Random samples from third cluster of three (enlarged city centers).

Chapter 6

Network Motifs

In this chapter, a different approach to finding the characteristics of cities is presented. It is based on network motifs, which are patterns that appear significantly often in a complex network. Compared to the clustering approach in chapter 5, this method focuses more on the spatial structure of the street network and less on purely statistical values. In the first part of this chapter, network motifs are defined and the methodology of motif detection is shown. The second section analyzes the detected motifs and the third section covers motif-based clustering.

6.1 Network Motif Detection

6.1.1 Previous work

The concept of network motifs and an algorithm for motif detection was first presented by Milo et al. [24]. Network motifs are therein defined as ”recurring, significant pattern of interconnections” in directed graph networks. Identified network motifs are thought to be linked with functional behaviour: A pattern which is found often in e.g. a neural network could encode a certain, basic functionality. While motifs were originally studied in gene regulation and other biological networks, motif detection can also be conducted on any other kind of complex networks. Milo et al. also showed that graph networks of different origins (such as neural networks, gene regulation networks or internet networks) contain distinct sets of significant network motifs, indicating that the underlying dynamics of these network types differ fundamentally.

Previously, Topirceanu et al. have already shown a comparative, motif-based street network analysis of six cities [25].

6.1.2 Algorithm

The basic motif detection algorithm works as explained below. First, the number of occurrences of all types of subgraphs are counted for the given network. Then, a large number (usually around 1 000) of random networks is generated and the counting process is repeated for all of them. These random networks are built such that their in- and outdegree statistics are identical to those of the original network. Furthermore, for motifs consisting of more than two nodes, the subgraph statistics are also enforced to be equal. Finally, the subgraph counts of the original network are compared to the average count of the random networks. If the probability of observing the measured frequency of a specific subgraph in the random networks is smaller than a given threshold, this subgraph is considered a detected motif.

Motif detection is computationally challenging for large networks as well as for motifs with more than three or four nodes. In this thesis, the algorithm needs to be run once for all cities, which means that subgraphs have to be counted in 251 city networks and 251 000 random networks of the same size. The street networks for big cities contain up to 270 000 nodes and 735 000 edges (Tokyo). In total, eleven city networks have more than 100 000 nodes each.

Besides that, the number of different n-node subgraphs increases largely with increasing n. For example, there exist only two 2-node subgraphs: Two nodes can be connected by either a unidirectional or a bidirectional edge. For $n = 3$, there are already 13 possible graphs, all shown in figure 6.1, and there exist 199 and over 9300 four- and five-node subgraphs, respectively. Note that only connected subgraphs are counted for these numbers.

The above-mentioned scaling effects limit the capabilities of the described algorithm. Motif finding is therefore generally considered unfeasible for motifs with more than about five or six nodes, which is the reason this study considers only motif with between two and five nodes.

Kashtan et al. provide a tool called `mfinder` for motif detection [26] which implements the algorithm of Milo et al. [24]. In a first attempt, `mfinder` was used for motif detection in the street networks. However, it would have taken multiple days to find the four-node motifs of just

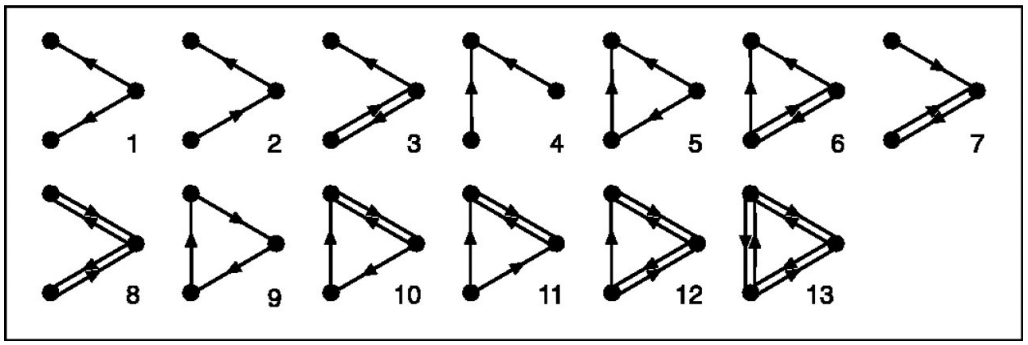

Figure 6.1: Illustration of all existing three node graphs. Figure taken from [24]

a single large city and probably weeks to conduct a full motif detection on all cities. An alternative is provided by the tool `FANMOD`, developed by Wernicke et al. [27]. It is based on an algorithm called `RAND-ESU` [28] and is much faster than `mfinder`. For large networks or motifs with more than about five nodes, it also offers a stochastic motif detection approach instead of a full enumeration of all subgraphs in the network. However, for the work presented in this thesis, only full enumeration was used. For one run where all three-node motifs in one of the largest cities were detected, `FANMOD` needed about two hours of computation time as opposed to more than two days for `mfinder`.

6.2 Motif Analysis

The resulting motif statistics were collected in four tables per city, one for two-node subgraphs, three-node subgraphs etc. `FANMOD` returns detailed statistics for all detected subgraphs including their frequency in both the given network and the random networks. It also indicates whether a specific subgraph appears significantly more often in the given network than the 1 000 random networks and therefore is a motif or not.

Additionally, the results for each subgraph size were combined in one single table. It lists for each subgraph the cities where it was detected an the respective frequencies of the subgraphs. Again, it also indicates the significance of the subgraphs. These data tables were the base for all further analyses and clustering applications.

For a first analysis, the number of detected significant motifs was plotted against the city population. For two-node motifs, this is not very conclusive, since there exist only two possible motifs. All cities must therefore have zero, one or two distinct, detected two-node motifs. No

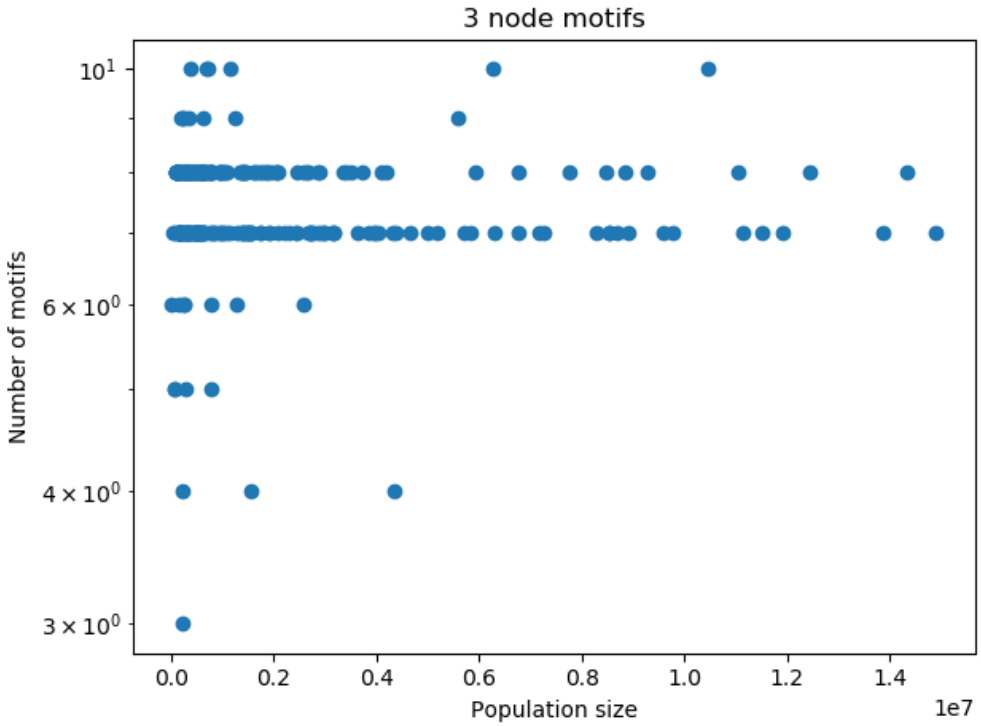

Figure 6.2: Population versus number of detected distinct three-node motifs.The y-axis is in log scale for better visualization.

clear trend or pattern could be found based on the results. Figures 6.2, 6.3 and 6.4 show the results for three-, four- and five-node motifs, respectively. In the three-node case we see that for large cities, there seems to be a trend in the number of motifs. While in small cities, very low or very very high numbers of distinct motifs were found, large cities tend to have a medium-to-high number of motifs. For larger motifs with four and five nodes, it seems that large cities tend to contain more distinct motifs than small cities.

Secondly, the total number of motif occurrences was counted and plotted against population. Unlike explained above, not distinct motifs were counted for each city. Instead, the absolute frequencies of all detected motifs are shown. Results for the most often detected three-node-motifs are depicted in figures 6.5 and 6.6. Motifs are assigned a unique ID based on their corresponding adjacency graph, hence the numbers 38 and 102. Note that cities where the corresponding subgraph was not detected as a significant motif were not plotted to avoid confusion. Furthermore, the number of occurrences of all detected motifs were summed up

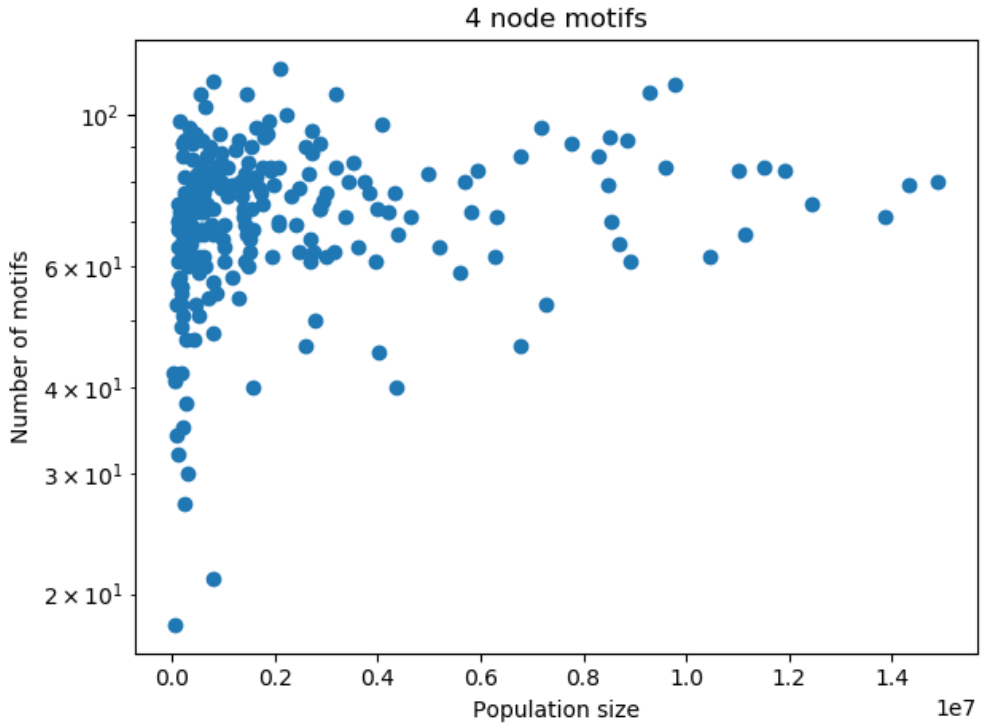

Figure 6.3: Population versus number of detected distinct four-node motifs.The y-axis is in log scale for better visualization.

for all cities. The resulting plot for three-node motifs is shown in figure 6.7, the corresponding plots for two-, four- and five-node motifs can be found in appendix B.

Different from the number of distinct motifs, there is no trend or visible difference between small and large cities for any motif size. The total number of motif occurrences per city does not correlate to city size at all according to this analysis. This implies that large cities do not consist of more repetitions of motifs, but instead of more diverse, insignificant subgraphs.

It is important to note that the city of Melbourne has been excluded from the plots above. Figure 6.8 shows an example where Melbourne is highlighted and obviously a strong outlier. It seems that Melbourne contains a huge number of motifs while its population is only in the lower medium range of all cities. The reasons for this could not be determined, although the city borders, the street network data and the population number were checked with precision. Melbourne's street

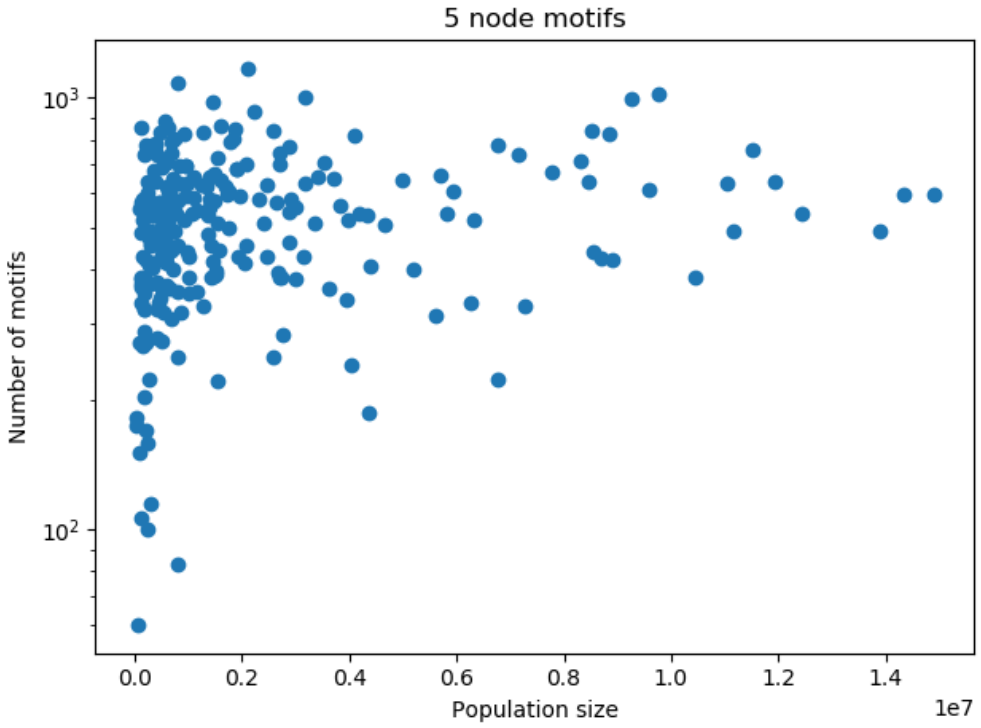

Figure 6.4: Population versus number of detected distinct five-node motifs. The y-axis is in log scale for better visualization.

network is not especially regular or shows any other peculiarity. One reason could however be the high number of sub-cities it has. For better visualization of the other cities, it was decided to remove Melbourne from the data set before plotting.

Furthermore, figure 6.9 shows all possible three-node motifs and the number in red indicates for how many cities this specific subgraph was detected as a significant motif. We see that some motifs were hardly detected at all (for example the motifs with ID 6 or 36) while others appear in nearly all cities (such as ID 38 or 238). However, most interesting are the motifs that appear in some cities, but not all. Motifs that appear significantly often in all cities might be fundamental building blocks of cities in general, but obviously their existence does not tell us anything about the properties of a city. On the other hand, the relevance of a motif that appears in only two cities at all is probably very low.

If we consider only three node motifs that were detected in more than 10% and less than 90% of cities, a single one remains, the motif with

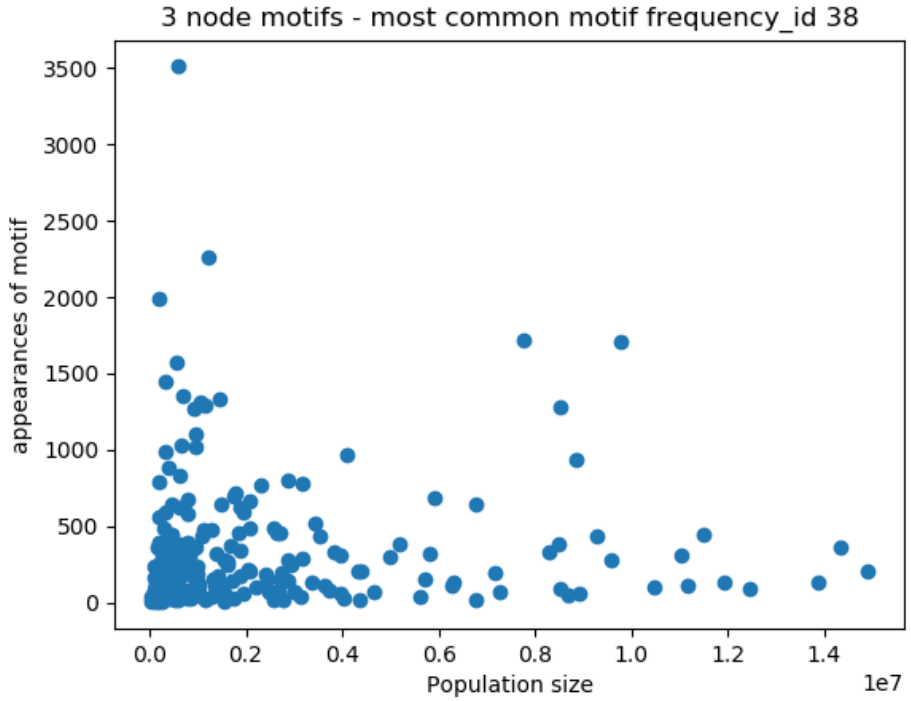

Figure 6.5: Population versus absolute frequency of three-node motif with ID 38. Only cities where motif 38 was detected were considered.

ID 78. It was detected in 105 of 251 cities and its statistical significance therefore splits the city data set in two parts of roughly similar size. Table 6.1 shows a complete list of the cities belonging to each cluster. When looking at the geographical distribution in figure 6.10, we clearly see that one cluster consists of cities from Europe, Australia and North America almost exclusively, while the other one contains African, Asian and South American Cities. In this clarity, this result is surprising and indicates that a further, deeper analysis of this and similar motifs is very promising.

6.3 Subgraph-based Clustering using LDA

In addition to statistical analyses of motifs, this section shows an approach to city clustering based on network subgraphs. It uses a tool from natural language processing, namely latent Dirichlet allocation (LDA).

Motif	Cities
Yes	Accra, Adelaide, Akureyri, Amsterdam, Antwerp, Basel, Belgrade, Bergamo, Bergen, Berlin, Bilbao, Birmingham, Bologna, Boston, Braga, Brasilia, Brest, Brisbane, Bristol, Brno, Brussels, Budapest, Calgary, Cali, Canberra, Coimbra, Copenhagen, Darwin, Dortmund, Dubai, Dublin, Fribourg, Galway, Geneva, Glasgow, Graz, Halifax, Hamburg, Helsinki, Isfahan, Karlsruhe, Katowice, Kiev, Krakow, Lausanne, Leeds, Leicester, Lisbon, Liverpool, London, Los Angeles, Lucerne, Lyon, Madrid, Malmo, Manchester, Marseille, Mashhad, Melbourne, Miami, Milan, Moscow, Munich, Naples, New York, Odense, Odessa, Oslo, Ostrava, Pamplona, Paris, Plovdiv, Porto, Posadas, Prague, Rennes, Reno, Reykjavik, Rome, Saint Petersburg, Salzburg, San Francisco, Seville, Sharjah, Sibiu, Singapore, Stavanger, Stockholm, Sydney, Timisoara, Toronto, Trier, Trondheim, Vancouver, Vantaa, Venice, Vienna, Vigo, Warsaw, Washington, Winnipeg, Wolfsburg, Yekaterinburg, York, Zurich
No	Abidjan, Addis Ababa, Aktau, Algiers, Almaty, Ansan, Aracaju, Arak, Arnhem, Asan, Athens, Atlanta, Austin, Baghdad, Bangalore, Bangkok, Barcelona, Beijing, Belem, Bellinzona, Bogota, Bucharest, Buenos Aires, Bursa, Cairo, Campinas, Cape Town, Caracas, Cartagena de Indias, Casablanca, Changwon, Charleroi, Chennai, Chicago, Chiclayo, Chittagong, Chongqing, Cork, Corrientes, Cuiaba, Das es Salaam, Daugavpils, Denpasar, Dhaka, Donetsk, Durban, Ecatepec de Morelos, El Alto, Feira de Santana, Fortaleza, Guangzhou, Gujrat, Hai Duong, Hanoi, Harare, Hirosaki, Ho Chi Minh City, Houston, Hyderabad, Islamabad, Izmir, Jakarta, Jinan, Johannesburg, Kabul, Kagoshima, Karachi, Kasama, Kharkiv, Khartoum, Kinshasa, Kyoto, Lagos, Las Vegas, Lima, Luanda, Maceio, Malaga, Maracaibo, Mariupol, Matsumoto, Medan, Medellin, Mexico City, Miskolc, Monterrey, Montreal, Mosul, Mumbai, Nairobi, Nanjing, New Delhi, Niigata, Nonthaburi, Novosibirsk, Osaka, Ottawa, Ouagadougou, Pereira, Philadelphia, Phoenix, Porto Alegre, Pretoria, Puente Alto, Quebec, Regina, Resistencia, Riga, Rio de Janeiro, Rosario, Rotterdam, Salt Lake City, Salta, Salvador, Santa Fe de la Vera Cruz, Santiago, Sao Luis, Sao Paulo, Seoul, Sevastopol, Shanghai, Shymkent, Sofia, Subotica, Suncheon, Surabaya, Talca, Tehran, Thessaloniki, Tokyo, Tunis, Ube, Udon Thani, Ulsan, Utrecht, Valencia, Vina del Mar, Vinh, Volgograd, Xiamen, Xining, Yakutsk, Yen Bai, Yokohama, Zahedan

Table 6.1: Full city list for both clusters, one where motif 78 is significant and the other one without significant motif 78.

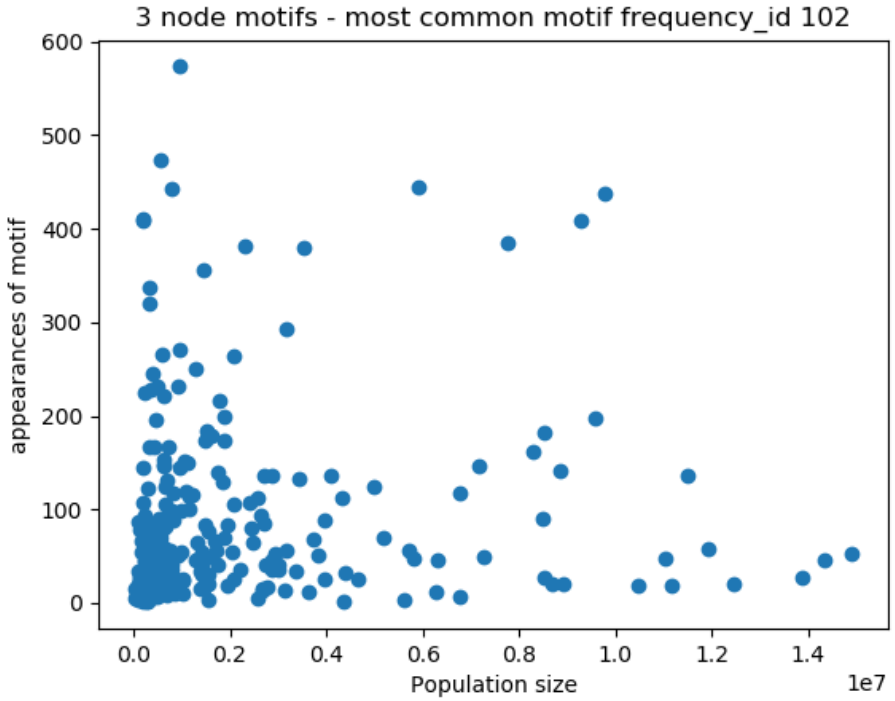

Figure 6.6: Population versus absolute frequency of three-node motif with ID 102. Only cities where motif 102 was detected were considered.

Latent Dirichlet allocation is a probabilistic modelling technique in natural language processing. Blei et al. [29] presented this model in 2003. Since then, various extensions have been developed [30], some of them able to use phrases instead of single words [31] [32].

In natural language processing, LDA works on a set of documents, each consisting of a large number of words. It is based on the so-called *bag of word* approach which assumes that the order of the words can be neglected. Therefore, the number of occurrences of each word is counted for all documents in the set. LDA then assumes the existence of k topics where k is known in advance. In every topic, each word has a particular probability. For example, if there exist the two topics *family* and *sports* the word *father* might have a high probability to appear in text generated from the *family* topic but low probability to be part of a *sports* text.

According to the LDA model, documents are generated in the following way: For each word in the document, a topic is drawn randomly at first. Then, using the word probabilities of the drawn topic, a word is

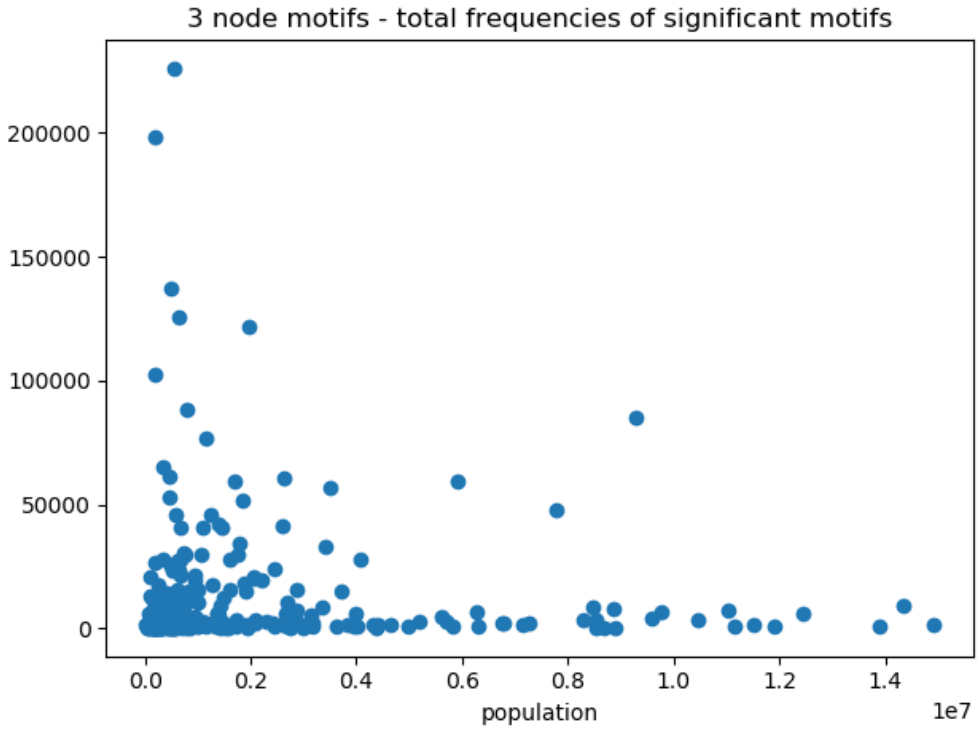

Figure 6.7: Population versus summed absolute frequency of detected three-node motifs.

randomly drawn. Note that for each word in the document, a new topic is drawn. A document therefore contains a mixture of multiple topics. LDA finds the most probable topic for each word in a document and can be used for dimensionality reduction. Instead of storing the full word counts of a document, it can be represented by its topic distribution (e.g. 20% *sports*, 80% *family*). For more detailed information about LDA, please refer to Blei et al. [29].

We can map this to the subgraphs detected in section 6.1. Each city corresponds to a document and each subgraph count to a word count. We therefore assume that cities consist of a mixture of city topics which generate parts of the street networks. More formally, we gather the results from subgraph detection for city i in a vector $\mathbf{x}_i$. Each component contains the counts of one subgraph.

$$\mathbf{x}^i = (\underbrace{c_1^2 \quad c_2^2}_{\substack{2-node\\subgraphs}} \underbrace{c_1^3 \quad \dots \quad c_{13}^3}_{\substack{3-node\\subgraphs}} \underbrace{c_1^4 \quad \dots \quad c_{199}^4}_{\substack{4-node\\subgraphs}} \underbrace{c_1^5 \dots \quad c_{9346}^5}_{\substack{5-node\\subgraphs}})^T \tag{6.1}$$

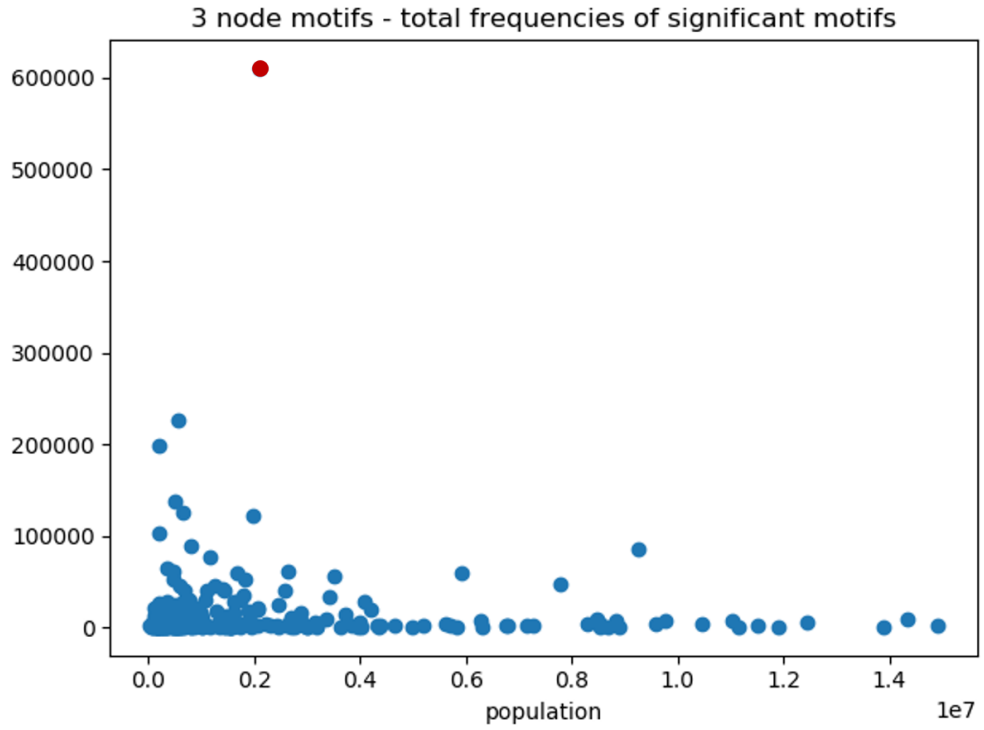

Figure 6.8: Population versus summed absolute frequency of detected three-node motifs, Melbourne included and highlighted in red.

Note that the information whether a particular subgraph occurs significantly often and is therefore a detected motif is not included in $\mathbf{x}$ and all subgraph counts, whether or not significant, are entered. One could easily define a diagonal matrix $\mathbf{S}^i$ with diagonal elements $s^i_{jj} \in 0, 1$, where the value 1 indicates that subgraph j is a motif and 0 indicates that subgraph j is not a motif. To get only the counts of significant subgraphs, i.e. motifs, use $\hat{\mathbf{x}}^i = \mathbf{S}^i\mathbf{x}^i$. However, large parts of a city might be lost if only motifs are considered, so it was decided to use the full subgraph counts. Only subgraphs that were never detected in any city from the data set were excluded.

Since the number of topics in the city data sets is not known, this number was set to $k = 50$. LDA therefore reduced the dimensionality of the data set from about 10 000 as shown in equation 6.1 to 50 dimensions. Unsupervised K-means clustering (see section 5.1) was then conducted in LDA-space. The results for $K = 2$ are shown in figures 6.11 - 6.14 below, the results for other numbers of clusters can be found in appendix

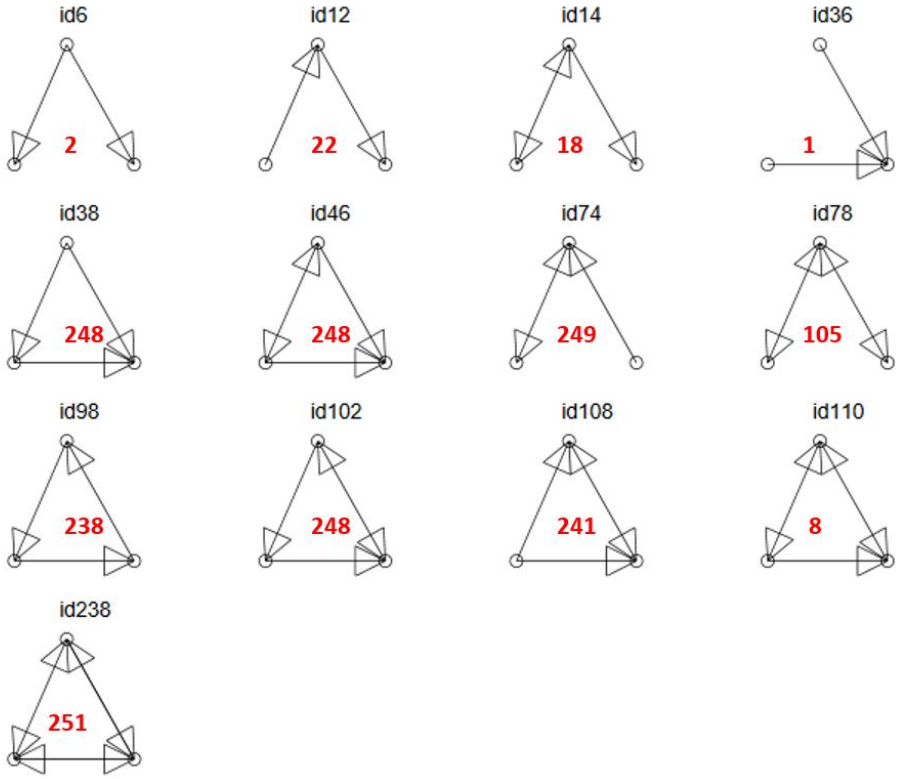

Figure 6.9: All possible three-node motifs. The number in red indicates the number of cities (out of 251) in which this graph was detected as a significant motif.

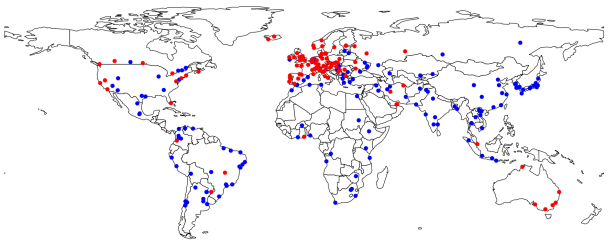

Figure 6.10: Geographical distribution of cities wherein motif 78 is significant (red) or not (blue).

C.

When looking at the city centers in figures 6.13 and 6.14, there is no fundamental difference between these clusters. It seems that the cities in the second cluster are somewhat more regular and grid-based, while the street networks in the first cluster follow a more organic pattern. However, Rome was assigned to the second cluster and does not really fit into this description. There is also no clear geographical pattern, the cities in the two clusters seem to be belong to all continents and countries. Similarly, this analysis is true for all clustering results in LDA space, not only the results for two clusters. Careful fine-tuning of LDA-based clustering might improve the results significantly.

Figure 6.11: Results of clustering based on LDA. Random samples for the first cluster of two.

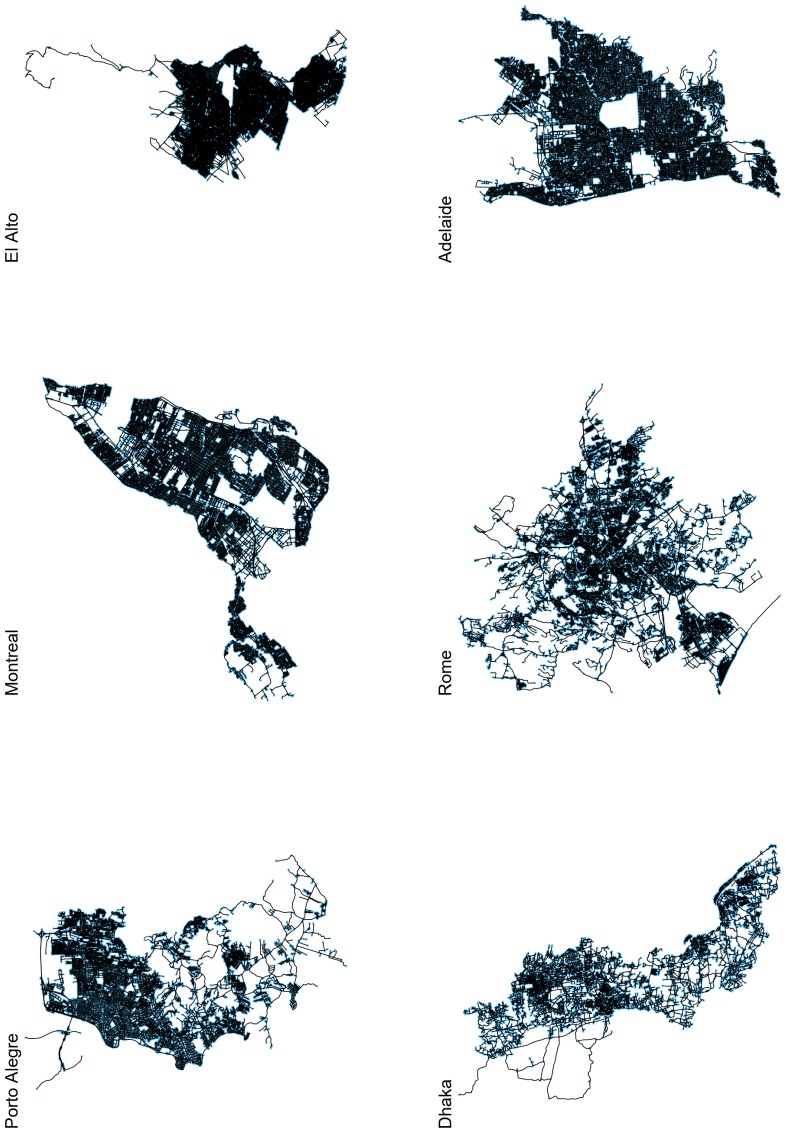

Figure 6.12: Results of clustering based on LDA. Random samples for the second cluster of two.

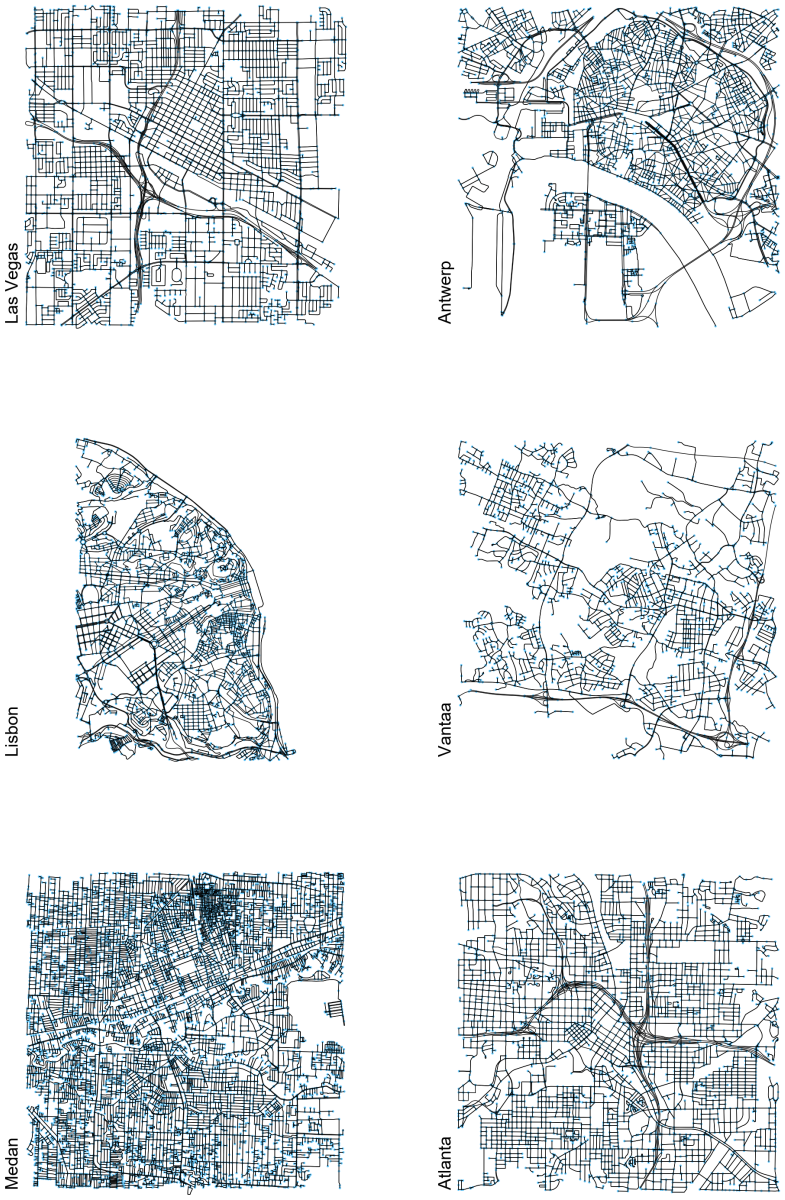

Figure 6.13: Results of clustering based on LDA. Random samples of city centers for the first cluster of two.

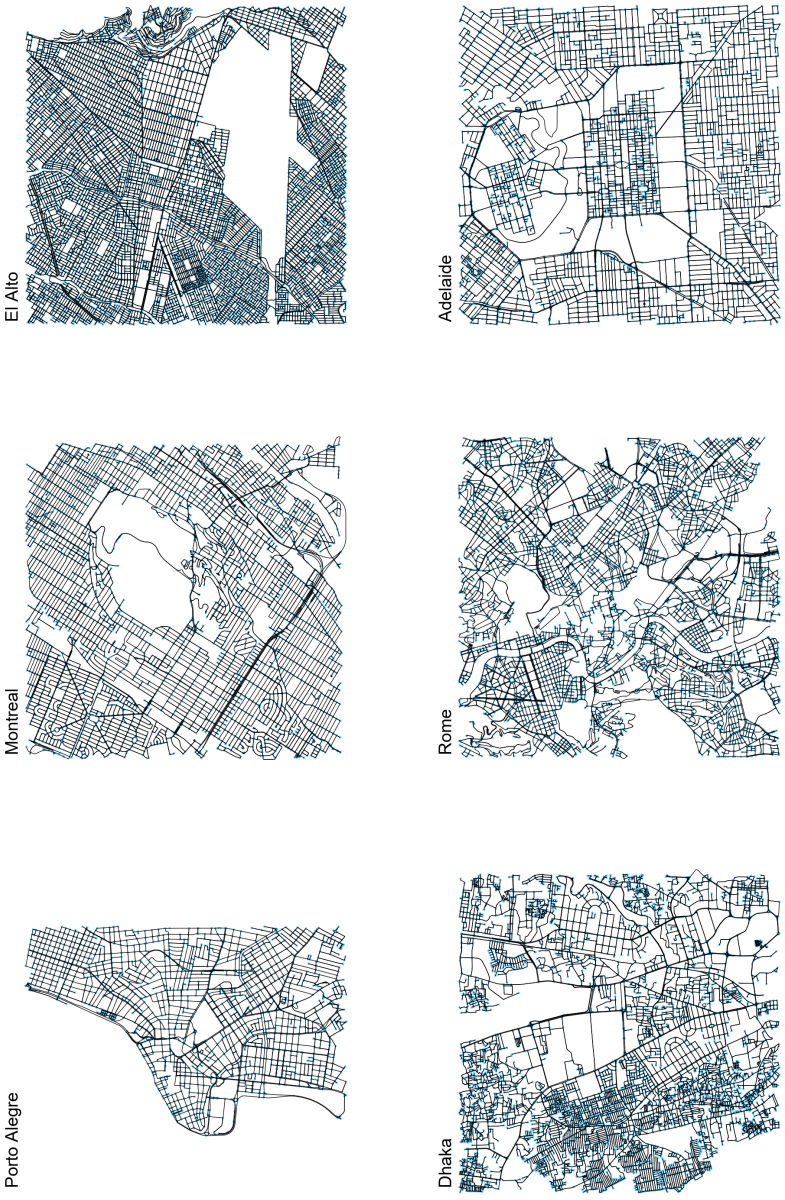

Figure 6.14: Results of clustering based on LDA. Random samples of city centers for the second cluster of two.

Chapter 7

Discussion

In this chapter, the data and results of this thesis are discussed. Future possibilities but also limitations and possible extensions are shown. The first section provides a closer look at the data while the results of the clustering approaches shown in chapters 5 and 6 are in the focus of the following two sections.

7.1 Data

The city data set of this thesis consists of 251 cities from all over the world, ranging from population numbers from below 200 000 up to over 13 million inhabitants. This high number of cities and the big variance in the set are sufficient for a sound statistical study. In chapter 4 we also analyzed correlations and found expected and already known correlations in the set as well as a variety of only weakly correlated features. This supports the claim that the data set is representative.

However, despite the efforts taken to ensure a balanced data set, big cities are still over-represented and some areas are badly covered, especially the middle east and most of Africa. Hopefully, in the next years, more city street networks will be available and updated regularly, especially from smaller and more rural cities and towns.

7.2 Clustering Approach

The results in section 5.3.2 show that it is possible to cluster cities based on relatively simple statistical measures from their street networks into

three groups which differ in city structure and size. This indicates that the street network statistics in fact contain usable information about city structure and the underlying building blocks of cities.

However, it is very important to note that both the methodology as well as the result are highly arbitrary. The reasons of choosing equation 5.2 for computing the total similarity have been discussed, but many other options are also possible. It would be particularly interesting to weight features individually by a scalar q^n for each feature n:

$$w_{ij}^{tot} = \sqrt{\sum_{n=1}^{25} q^n (w_{ij}^n)^2} \tag{7.1}$$

Using equation 7.1, the influence of some features with high weights is reinforced while the influence of others with low weights is mitigated. Such analysis might tell us more about the underlying structure, the importance of certain features and their coupling. It is very unclear how to choose such weights. Techniques from the fields of machine learning and optimization are capable of learning such parameters, but for this approach we would need some optimization goal, an objective or fitness function - basically a desired result. If we define the desired outcome in advance, however, we strongly limit the potential findings. The weights will only represent the importance of a certain feature for this specific, beforehand chosen outcome but not tell us the fundamental structural city elements.

This directly leads to the second arbitrary part: The result. The three clusters extracted by spectral clustering show distinct features and seem to represent three classes of cities well. But the resulting clusters would change if the computation of the total similarity (equation 5.2) was altered in any way, or if we would add or omit any features. The result is not stable in the sense that these clusters are based on fundamental underlying core characteristics, but it is highly dependent on the provided features and the methodology. Many other *reasonable* partitions of the data set are imaginable, for example one cluster containing all cities with a circular structure or a cluster with all cities that are built next to a river. There is no reason to assume that the partition shown above is more important or valid than any other partition.

Therefore, we must conclude that this approach, while it yields interesting results, is not suitable for the extraction of fundamental characteristics of cities.

7.3 Network Motif Approach

Chapter 6 has presented a completely different approach that is based on network motifs. The analysis shows that the number of detected distinct motifs follows somewhat a saturation pattern for large cities: When increasing the population, the number of motifs also increases and seems to converge. However, in the three-node motif case, small cities can either have very few or a very high number of motifs while both extremes do not occur for large cities.

Furthermore, it was shown that some motifs appear in significant parts of the data set, but not in all of them. This might indicate that their occurrence is connected to some urban features and that these motifs might correspond to some of the fundamental building blocks that we are looking for. However, more in-depth analysis is needed to fully understand the implications and verify the significance of these observations.

The motif statistics where then transformed via LDA and clustering was conducted in LDA space. However, the results were not fully convincing. Although it seems that some tendencies in the resulting clusters can be detected, there is a lot of noise.

We suppose that the problem lies in the basic assumptions of LDA, which treats documents as bags of words. However, maybe more than text in natural language, cities are spatial entities. The arrangement of subgraphs and not only their pure counts is crucial for the overall urban structure. Furthermore, subgraphs themselves are spatial and have a hierarchical ordering: A three-node graph consists of two or three two-node subgraphs. By counting all two-node subgraphs and all three-node subgraphs, we in fact count each edge at least twice, and the statistics of motifs of different sizes are not independent.

This difficulty does not arise in language processing, since words are distinct, isolated items. We would not detect the word *bad* in the word *badminton*, since words never overlap and must not be split. Given the full city network, it is however highly unclear where one subgraph ends and another one begins and what subgraph size has to be considered, and so the algorithm counts each part of each motif repeatedly.

As already mentioned in section 6.3, extensions for LDA that take groups of two or more words into account have been developed in the last years. Using such extensions for this thesis has been considered. However, language is a one-dimensional stream of words, where each word has a unique predecessor and successor. These can simply be grouped to cre-

ate phrases. In a two-dimensional irregular grid, on the contrary, there is no straight-forward way how to define neighboring or nested subgraphs in order to form *motif phrases*. Time and resources did not allow to dive deeper into the possibilities of phrase-based LDA clustering in the scope of this thesis.

For future work, we propose therefore a slightly different method where the subgraph statistics given so far can be regarded as (empirical) probabilities. Consider a hypothetical city which consists of only two different kinds of three-node subgraphs and let us denote them by their IDs 1 and 2. Assume that subgraph 1 represents 30% of the total number of three-node subgraphs while the remaining 70% consist of subgraph 2. The probability $P_3(1)$ of drawing subgraph 1 when choosing one three-node subgraph at random from the network is then equal to 30% or 0.3.

Additionally, conditional probabilities can be computed from subgraph statistics: The probability $P_{3|2}(y|x)$ denotes the conditional probability that a given two-node graph x is part of a three-node graph y. Since we know the composition and the frequencies of all three-node graphs of the city network, we simply need to count the number of times in which the two-node graph x occurs in any three-node graph and relate this to the number of times y appears. Similarly, we can compute empirical conditional probabilities for four- and five-node subgraphs.

One advantage of these conditional probabilities is that they encode much more spatial information than simple subgraph counts. Not only are the pure subgraph frequencies considered, but also the relationship towards neighboring subgraphs. Secondly, they allow to build a generative city model. Given all conditional probabilities of a specific city and a random two-node graph, we can extend it by repeatedly drawing new edges (considering the conditional probabilities) and adding them to the existing network until we reach the desired total network size. The result will clearly not be identical to the city it is based on, but have the same subgraph statistics and similar patterns. Simply put, we could build and explore for example a *Washington-like* city.

However, further research and time is needed to implement this only coarsely explained idea. Overall, although the results of our very naïve motif-based approach were not fully convincing, we see a lot of potential in the analysis of urban network subgraphs and various ways for extensions. Unlike basic features, network subgraphs also consider the spatial features of street networks and studying them for the goal of identifying the fundamentals of city structure seems very promising.

Chapter 8

Conclusion

The aim of this thesis was to explore two different approaches towards identifying the fundamental building blocks of city structure. A city model based on these building blocks could explain the properties of existing cities and also provide predictions for their future evolution.

First, a data base consisting of complete street networks of 251 cities from six continents was set up. A wide range of statistical measures of these cities were computed and complemented with population data mainly from the United Nations Statistics Division.

Subsequently, unsupervised clustering was applied to similarity graphs that were constructed from scalar and histogram features. The results show that it is possible to detect clusters with clear structural differences. However, as was explained in detail, we discovered two major shortcomings in this approach. On one hand, it is highly arbitrary and depends strongly on the input features and on the other hand, it is not suitable to identify basic building blocks since there is no importance weight on the features at all.

Secondly, a network-based approach was presented. Network motif detection was run on all cities of the data set and the resulting motif and subgraph statistics were again analyzed and used for clustering after the application of LDA. Some subgraphs were observed that occurred in only parts of the data set and might be candidates for the sought building blocks. Clustering based on LDA-transformed subgraph statistics results in groups of cities that also seem to share some common structural features.

Based on the results of this thesis, future research should concentrate on

motif-based analysis. Multiple fields for further exploration have been sketched in the last part of this thesis. Especially, deeper analysis of the resulting LDA clusters as well as a closer look at the motifs that appear in some but not all cities seems promising. Finally, implementing and expanding the conditional probability approach might eventually also lead to the development of a powerful generative city model.

Appendix A

Spectral Clustering Results

In the following sections, all spectral clustering results for $k = 2, 4, 5$ are shown. Details about the methodology are presented in chapter 5, where also the results for $k = 3$ are included.

A.1 Two Clusters

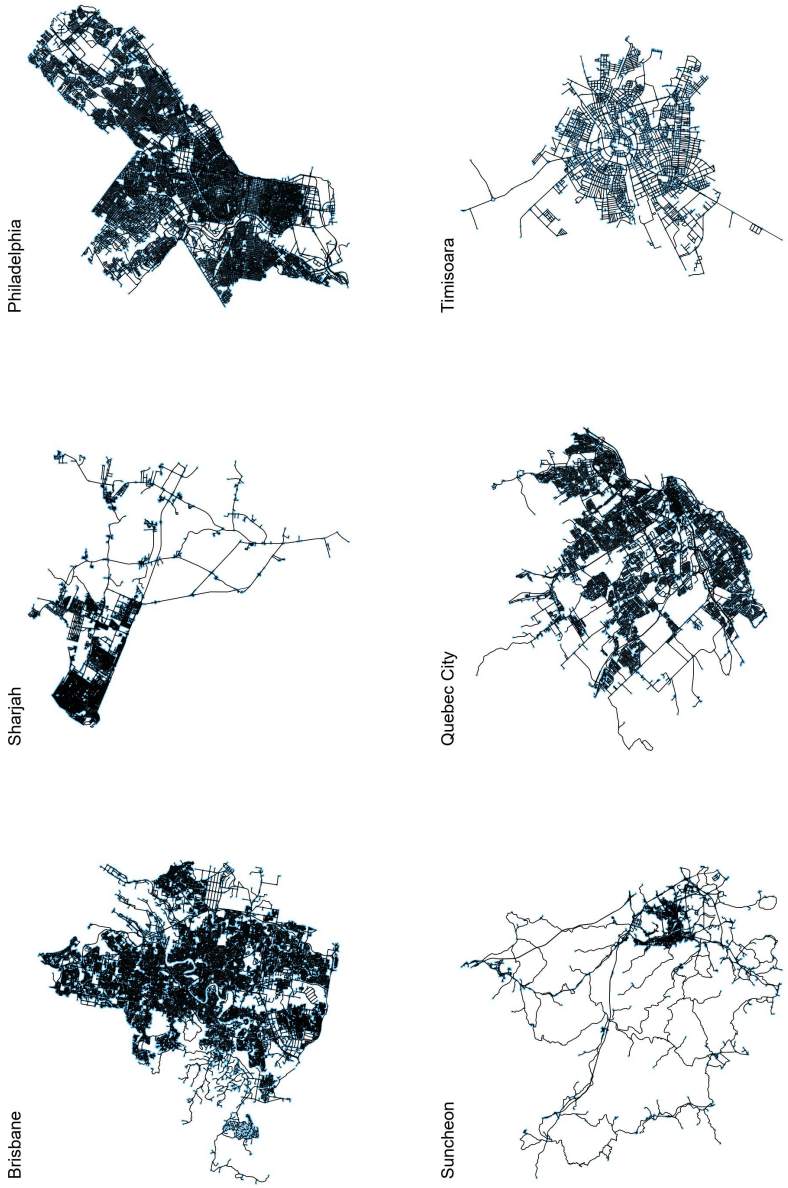

Figure A.1: Random samples from the first of two clusters resulting from spectral clustering.

Figure A.2: Random samples from the second of two clusters resulting from spectral clustering.

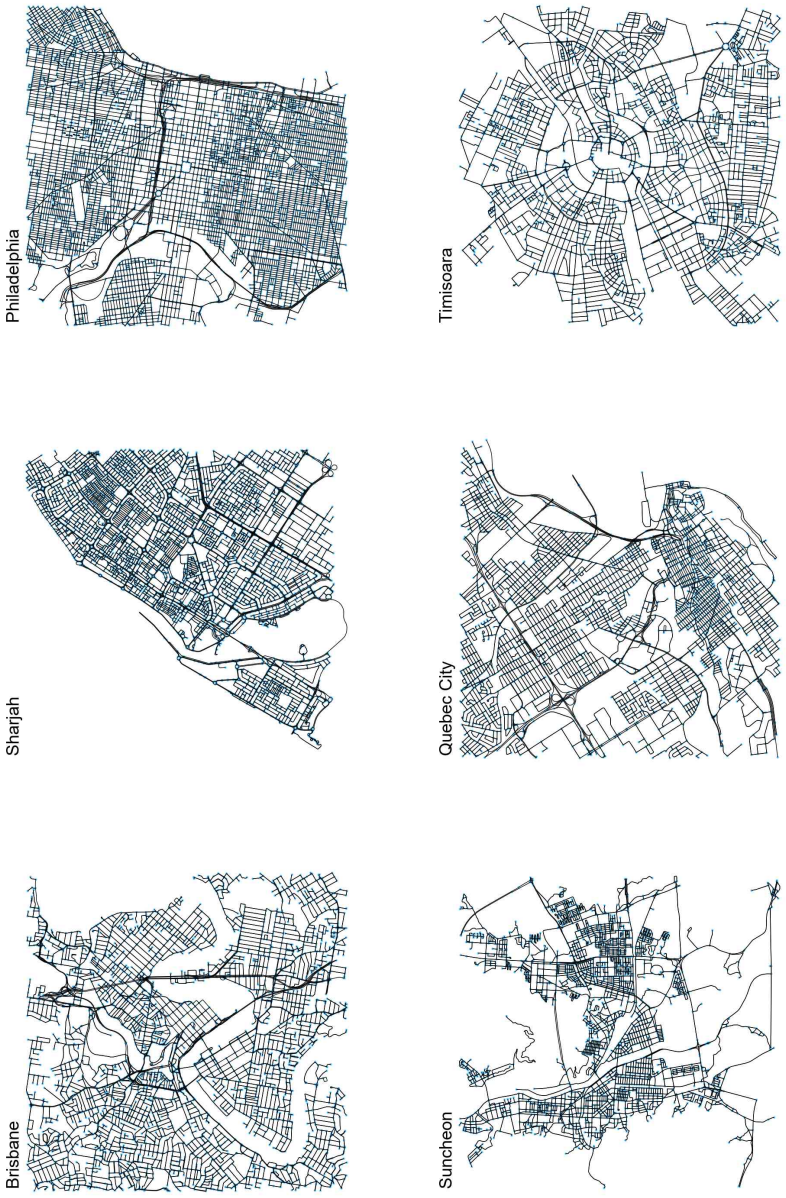

Figure A.3: Random samples from the first of two clusters resulting from spectral clustering (city centers only).

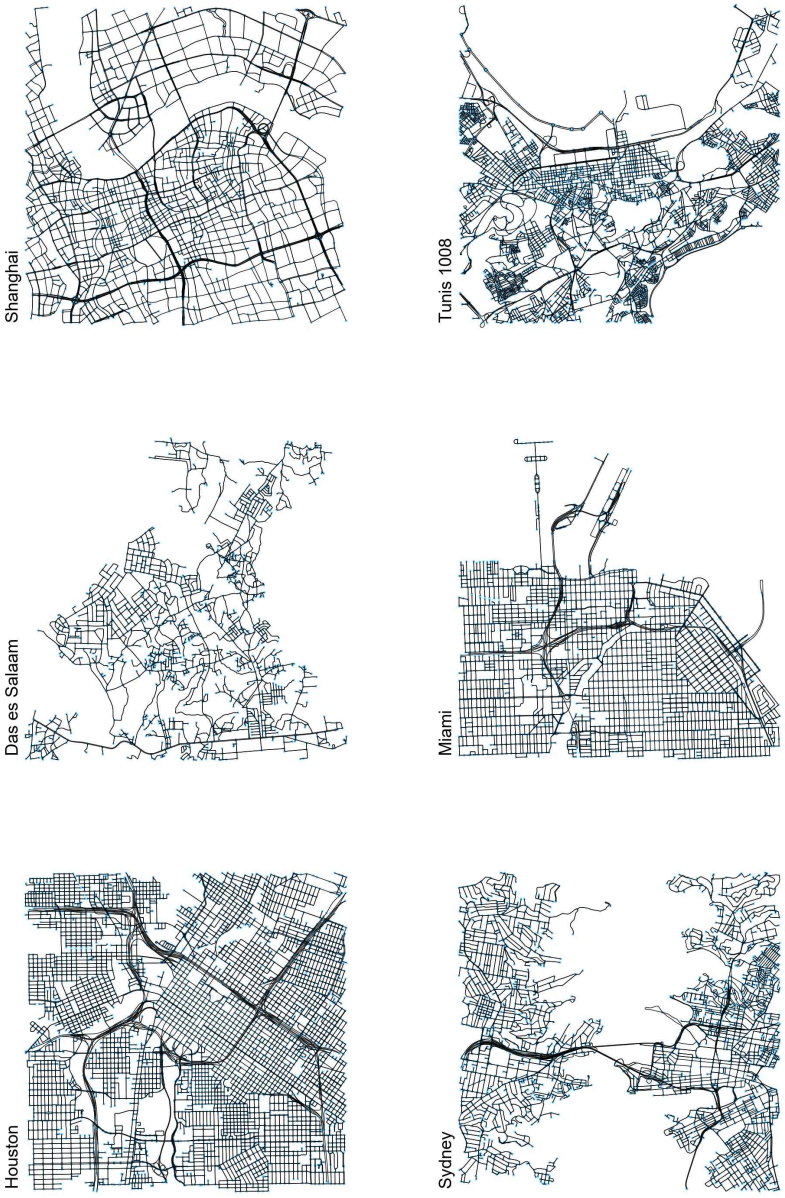

Figure A.4: Random samples from the second of two clusters resulting from spectral clustering (city centers only).

A.2 Four Clusters

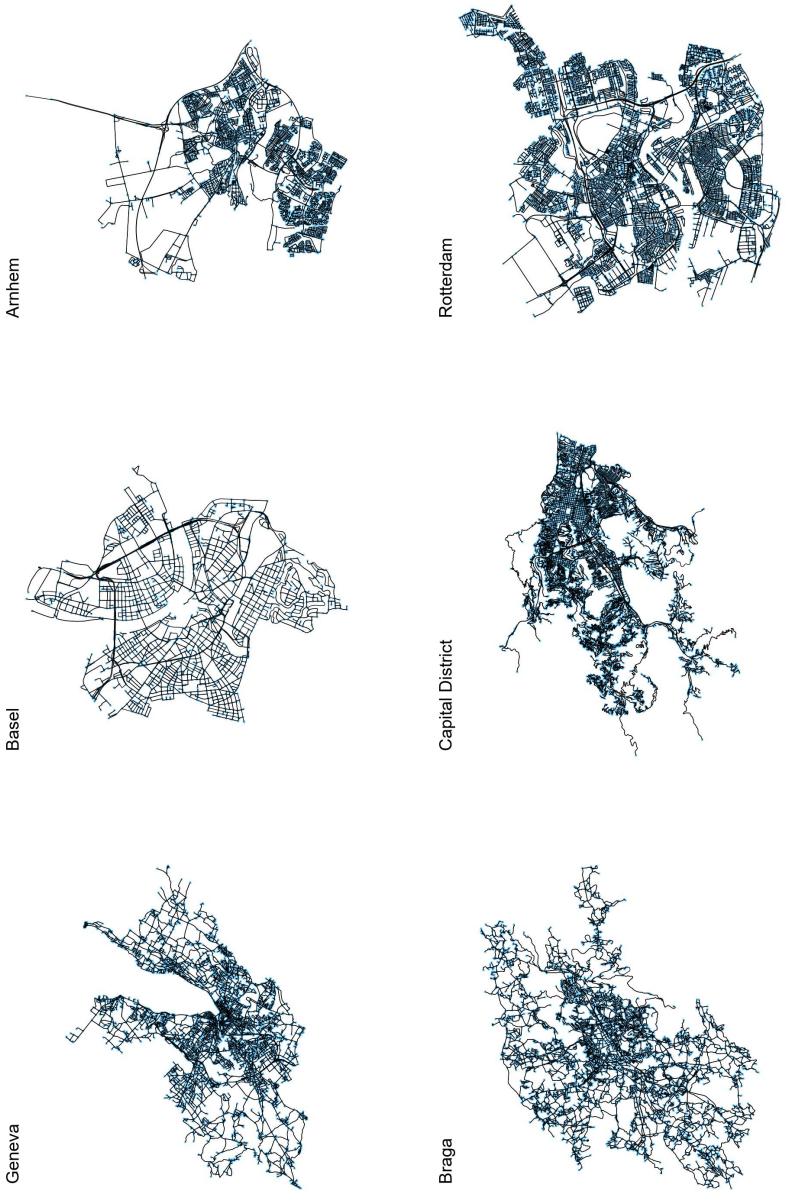

Figure A.5: Random samples from the first of four clusters resulting from spectral clustering.

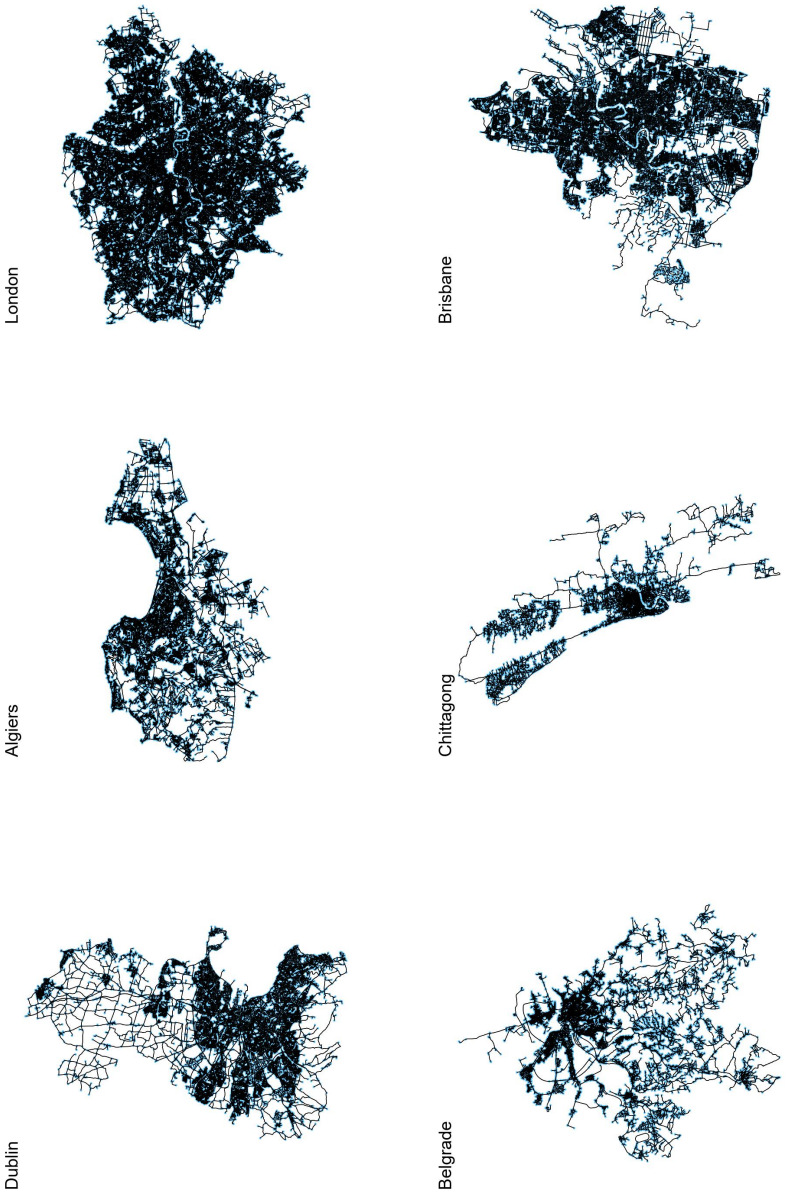

Figure A.6: Random samples from the second of four clusters resulting from spectral clustering.

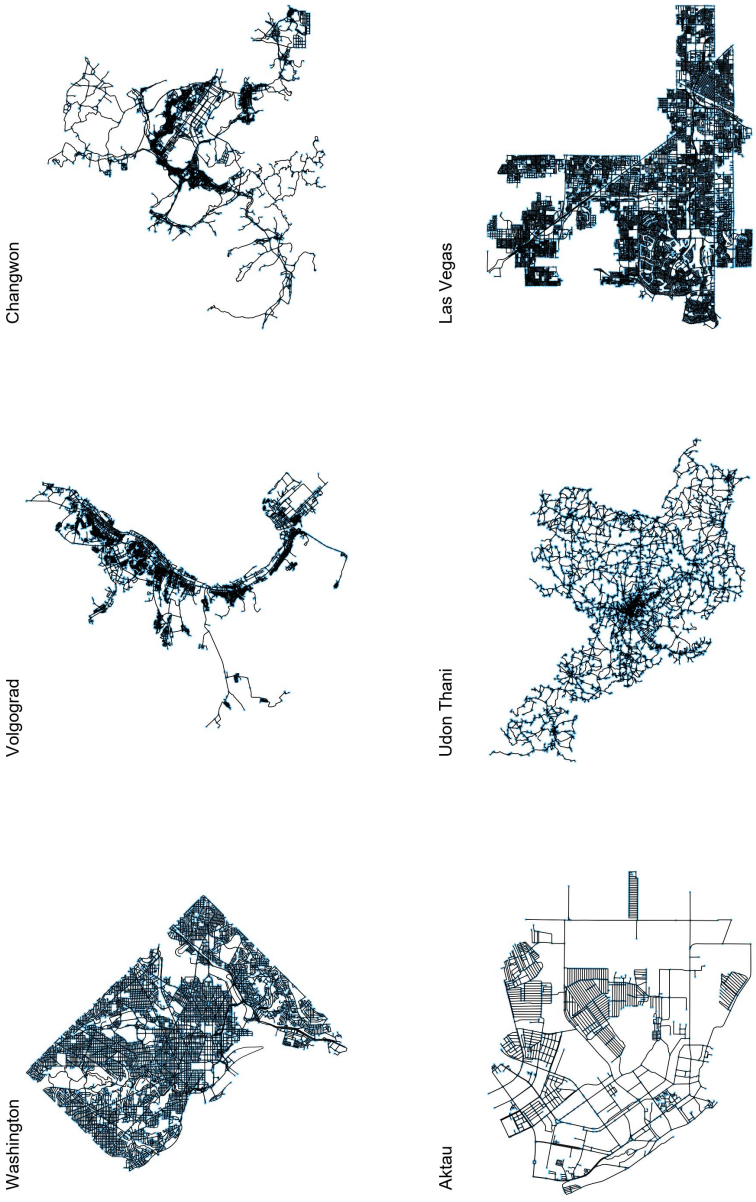

Figure A.7: Random samples from the third of four clusters resulting from spectral clustering.

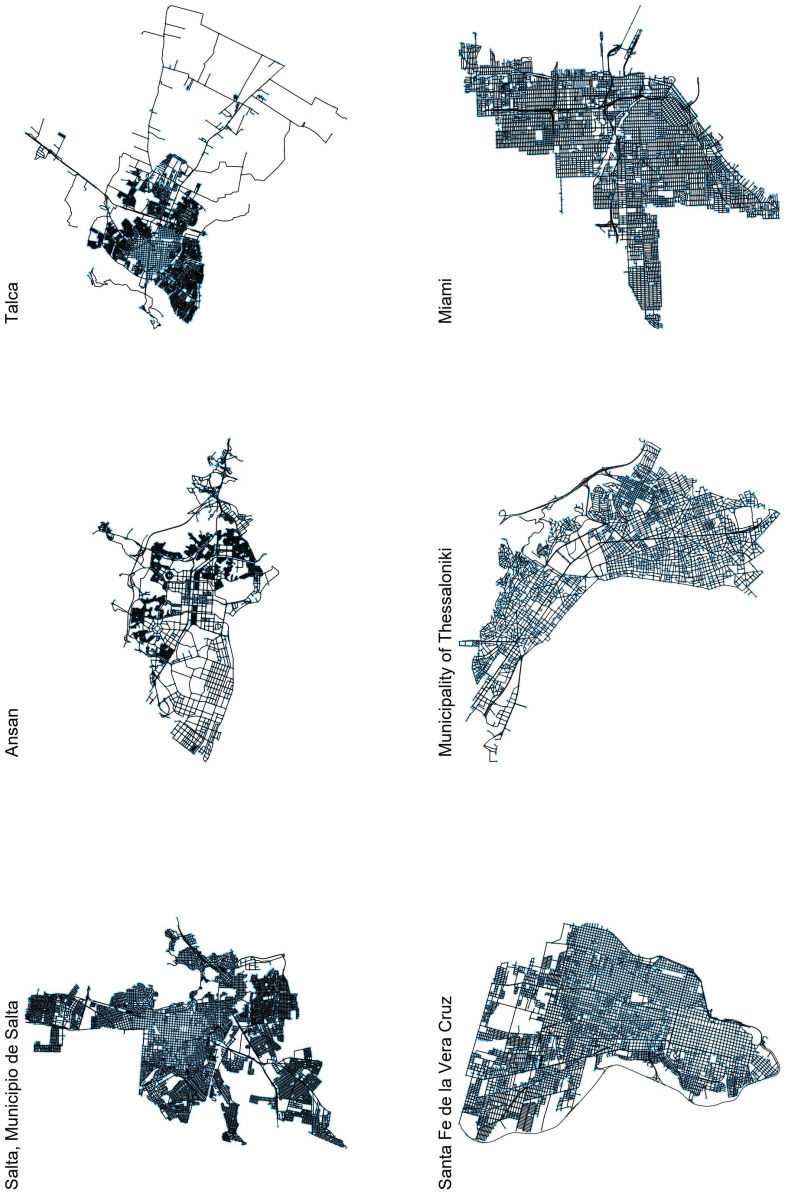

Figure A.8: Random samples from the fourth of four clusters resulting from spectral clustering.

Figure A.9: Random samples from the first of four clusters resulting from spectral clustering(city centers only).

Figure A.10: Random samples from the second of four clusters resulting from spectral clustering(city centers only).

Figure A.11: Random samples from the third of four clusters resulting from spectral clustering (city centers only).

Figure A.12: Random samples from the fourth of four clusters resulting from spectral clustering (city centers only).

A.3 Five Clusters

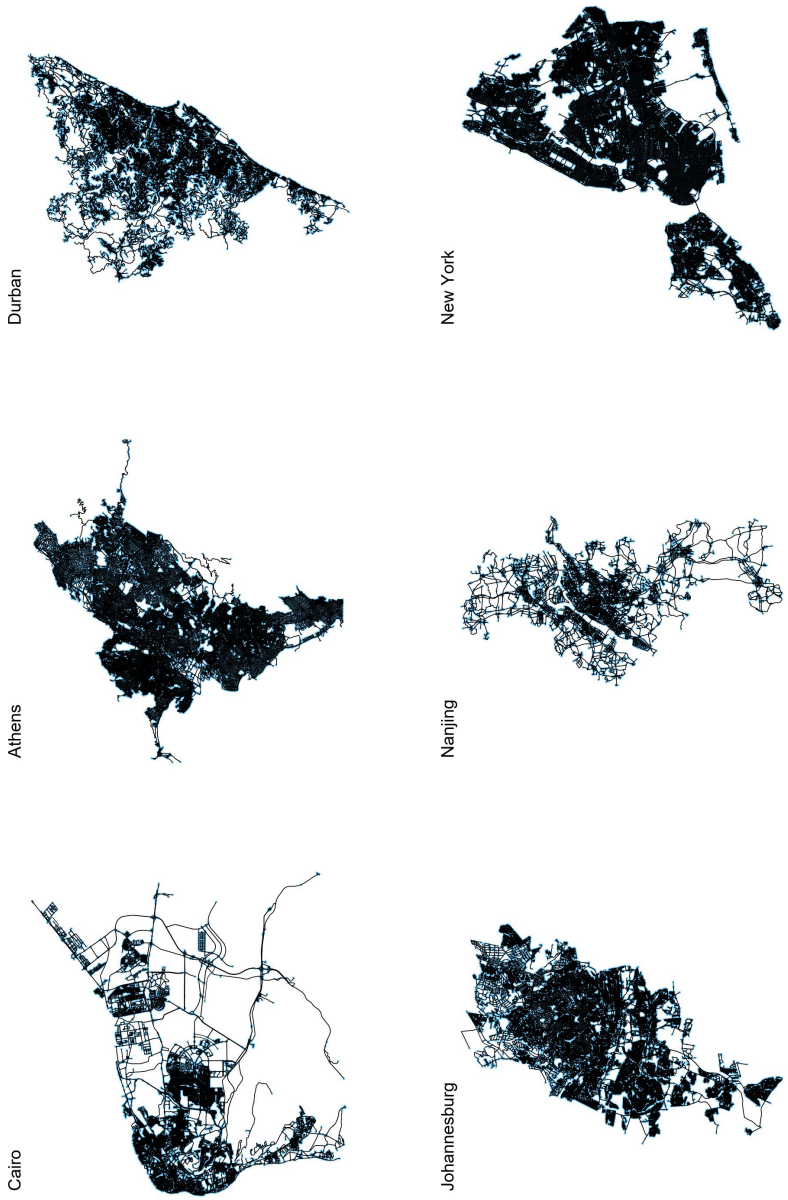

Figure A.13: Random samples from the first of five clusters resulting from spectral clustering.

Figure A.14: Random samples from the second of five clusters resulting from spectral clustering.

Figure A.15: Random samples from the third of five clusters resulting from spectral clustering.

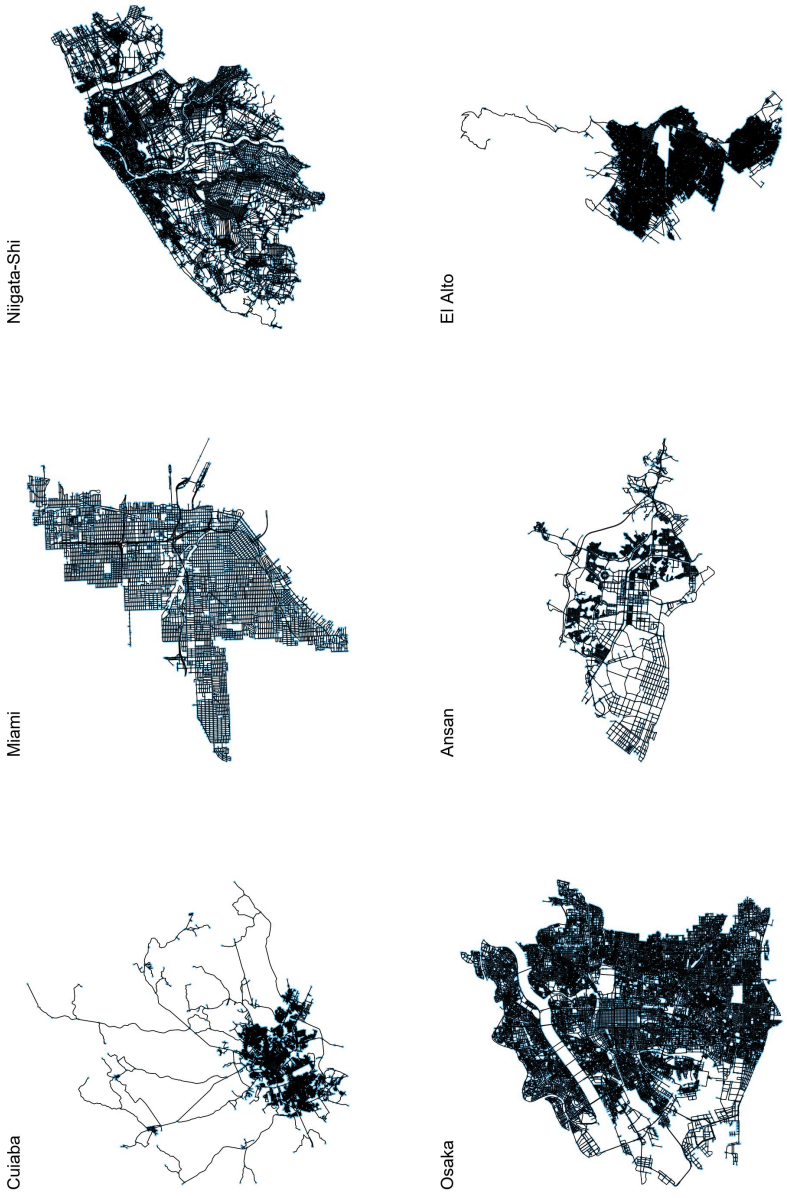

Figure A.16: Random samples from the fourth of five clusters resulting from spectral clustering.

Figure A.17: Random samples from the fifth of five clusters resulting from spectral clustering.

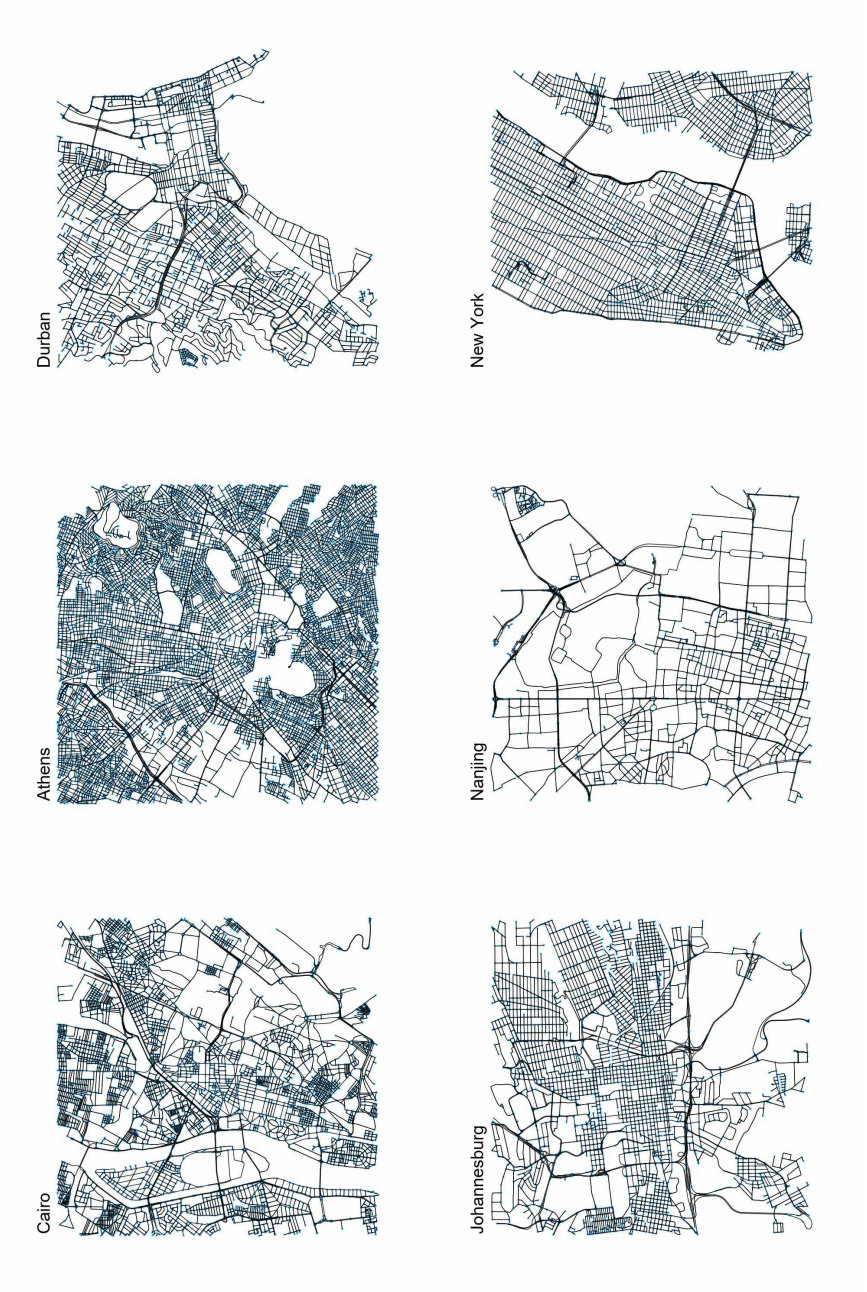

Figure A.18: Random samples from the first of five clusters resulting from spectral clustering (city centers only).

Figure A.19: Random samples from the second of five clusters resulting from spectral clustering (city centers only).

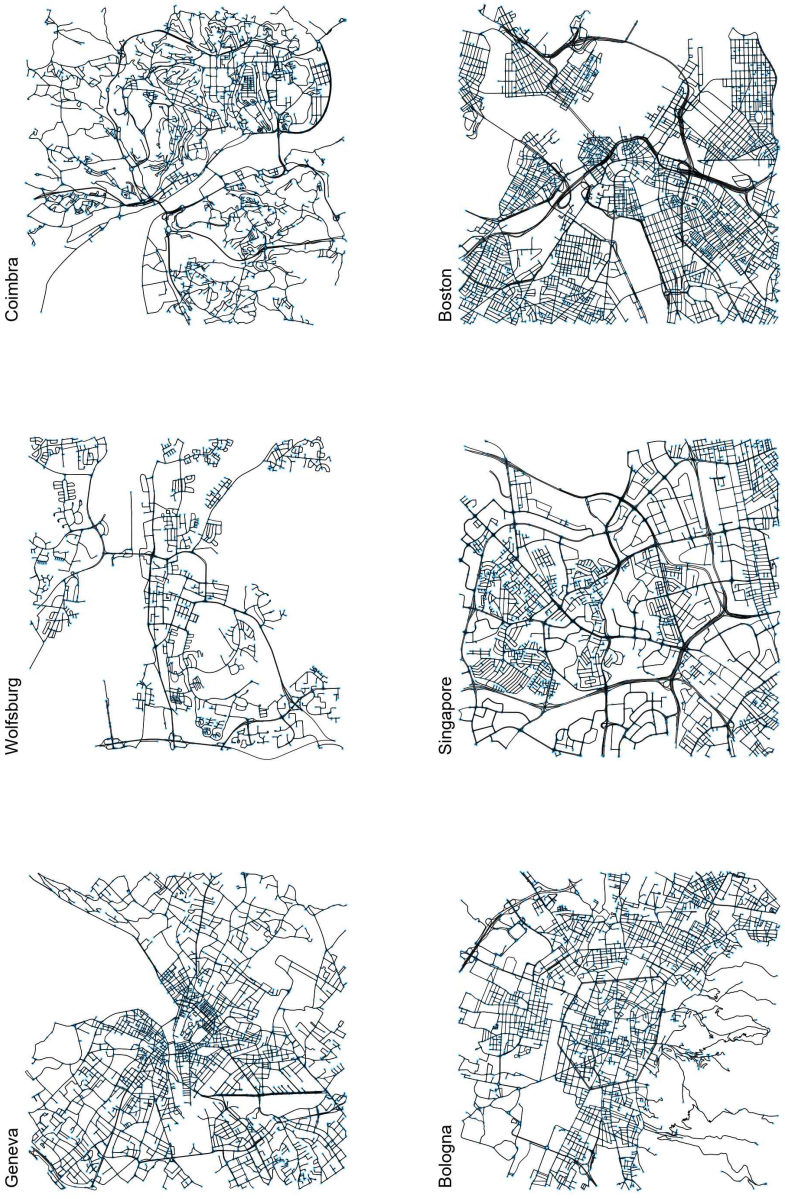

Figure A.20: Random samples from the third of five clusters resulting from spectral clustering (city centers only).

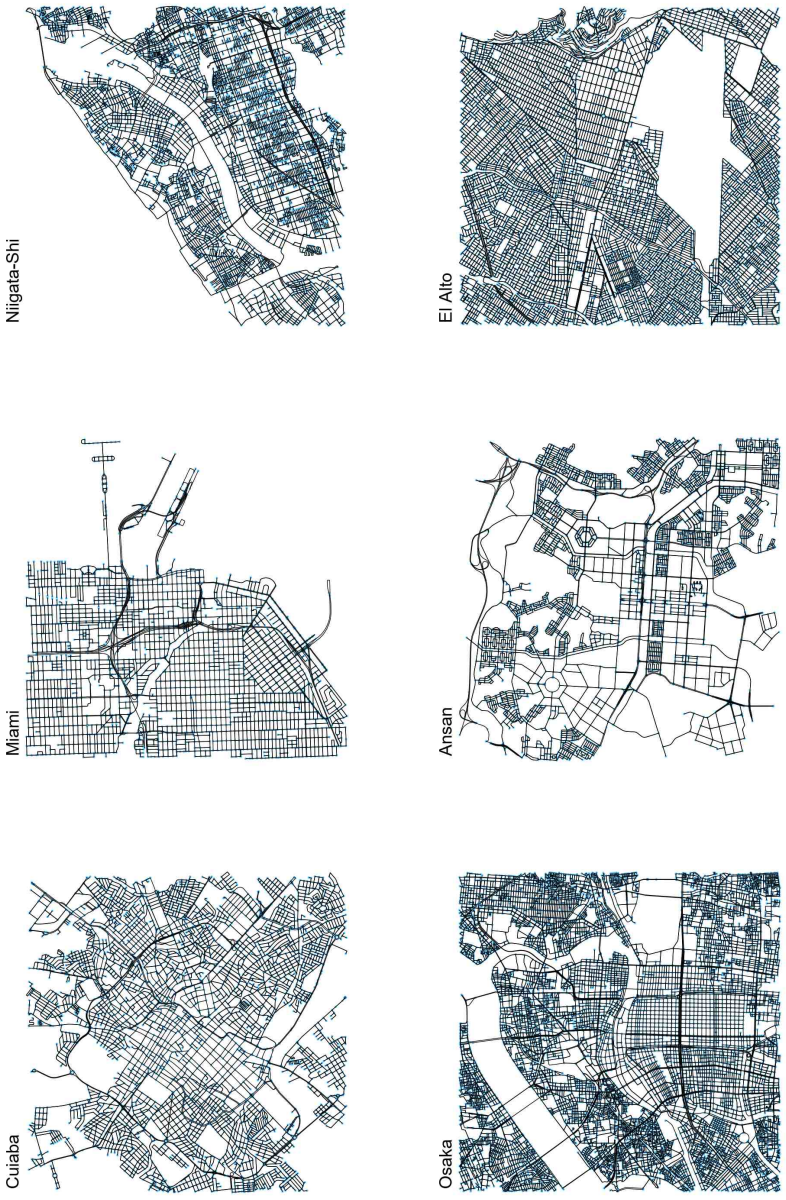

Figure A.21: Random samples from the fourth of five clusters resulting from spectral clustering (city centers only).

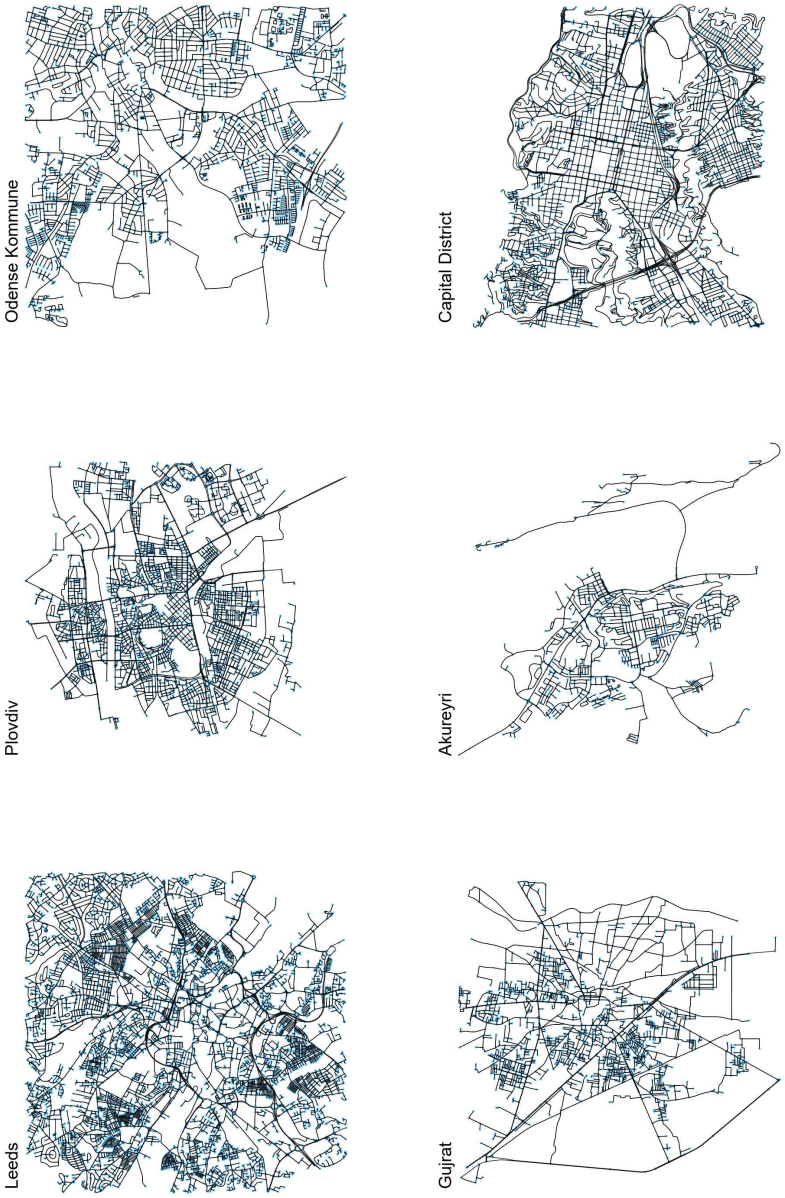

Figure A.22: Random samples from the fifth of five clusters resulting from spectral clustering (city centers only).

Appendix B

Motif Frequencies

This chapter shows the motif frequency results that were not discussed in detail in chapter 6. All of the data shown below was generated using the tool FANMOD.

For convenience, a console version of FANMOD provided on github [33] was run on the ETH Euler cluster. In advance, a python script was written to save the street networks in edge list format such that FANMOD could parse it. A simple bash-script was written on Euler in order to loop over all cities and call FANMOD for each city and each motif node number between two and five, each time generating 1 000 random networks for comparison.

B.1 Two-node Motifs

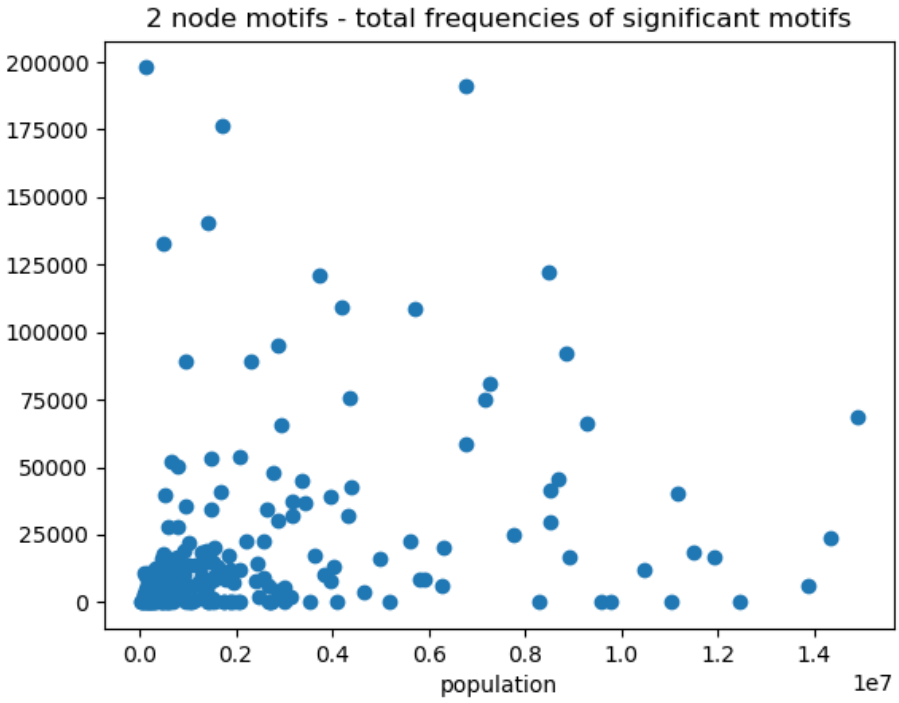

Figure B.1: Population versus summed absolute frequency of detected 2-node motifs.

B.2 Four-node Motifs

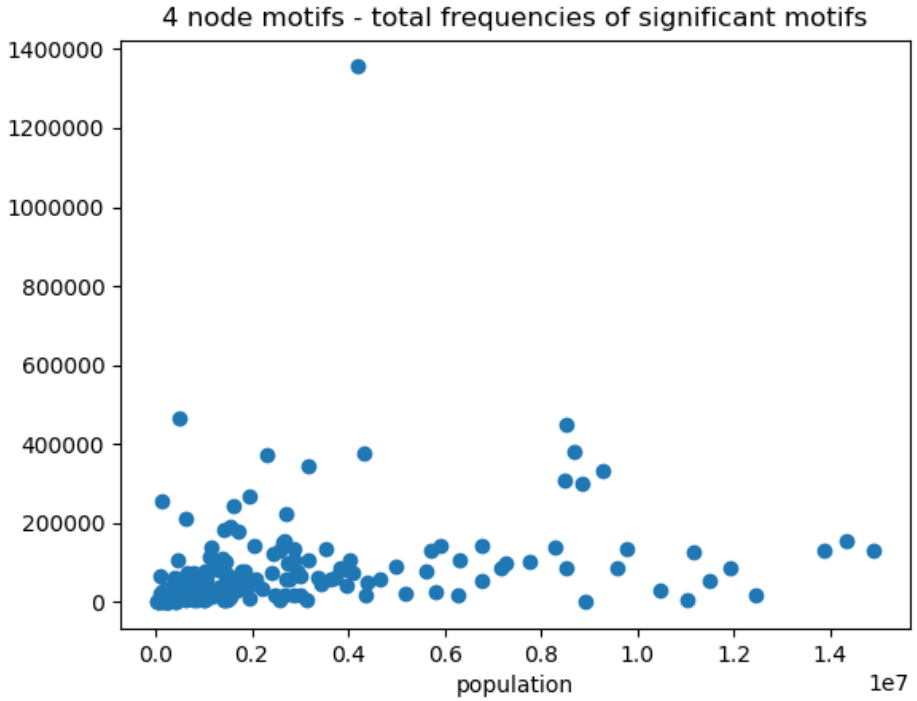

Figure B.2: Population versus summed absolute frequency of detected 4-node motifs.

B.3 Five-node Motifs

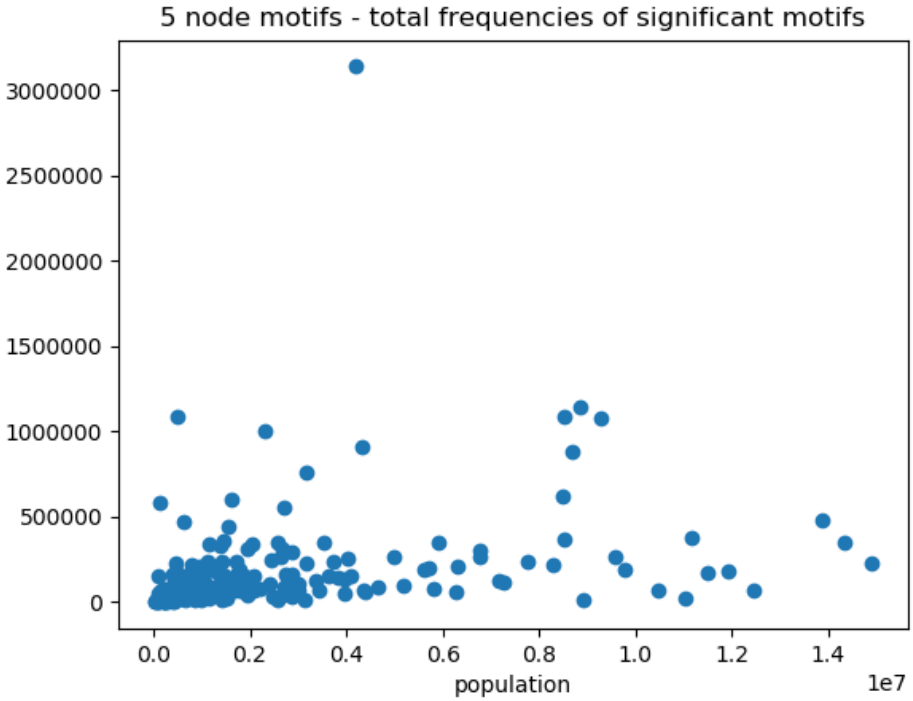

Figure B.3: Population versus summed absolute frequency of detected 5-node motifs.

Appendix C

LDA Clustering Results

This chapter shows all results from K-means clustering on LDA data. Not included are the results for $k = 2$ clusters, which can be found in chapter 6.

C.1 Three Clusters

Figure C.1: Random samples from the first of three clusters resulting from clustering in LDA space.

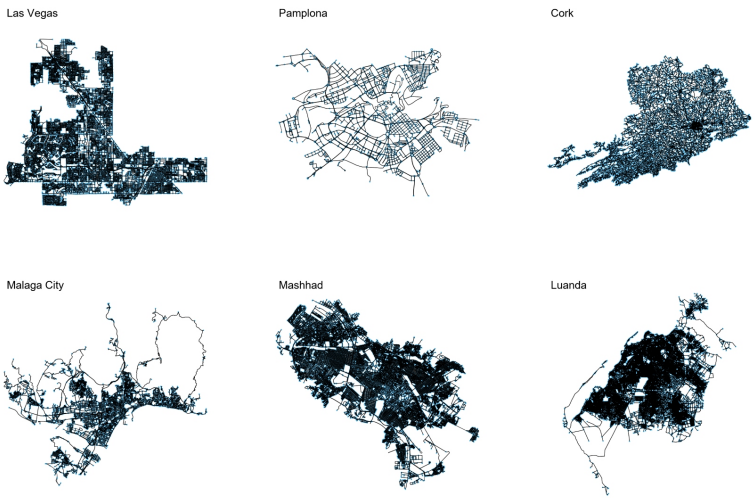

Figure C.2: Random samples from the second of three clusters resulting from clustering in LDA space.

Figure C.3: Random samples from the third of three clusters resulting from clustering in LDA space.

C.2 Four Clusters

Figure C.4: Random samples from the first of four clusters resulting from clustering in LDA space.

C.3 Five Clusters

Figure C.5: Random samples from the second of four clusters resulting from clustering in LDA space.

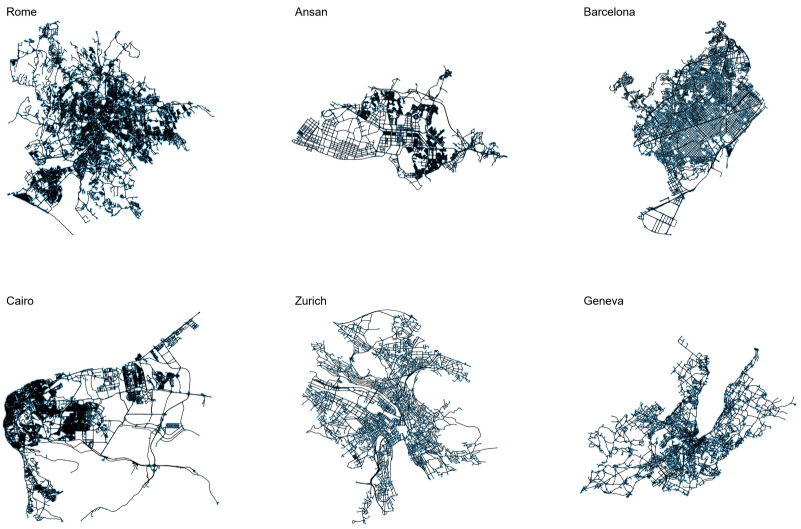

Figure C.6: Random samples from the third of four clusters resulting from clustering in LDA space.

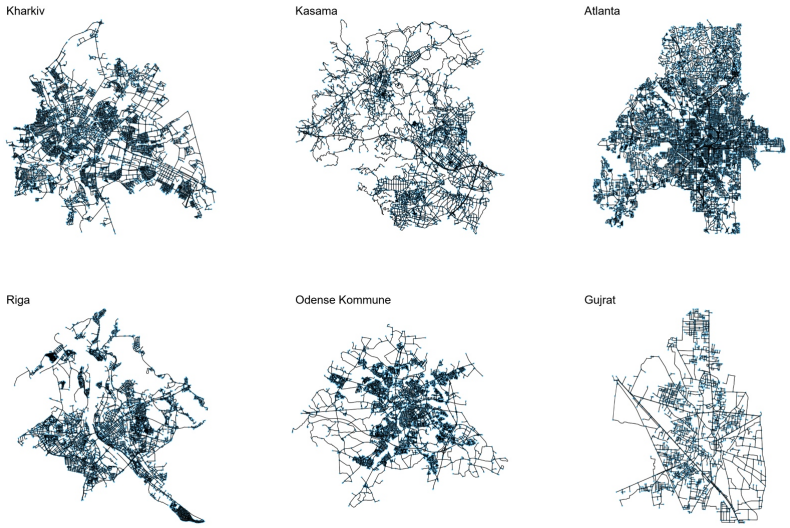

Figure C.7: Random samples from the fourth of four clusters resulting from clustering in LDA space.

Figure C.8: Random samples from the first of five clusters resulting from clustering in LDA space.

Figure C.9: Random samples from the second of five clusters resulting from clustering in LDA space.

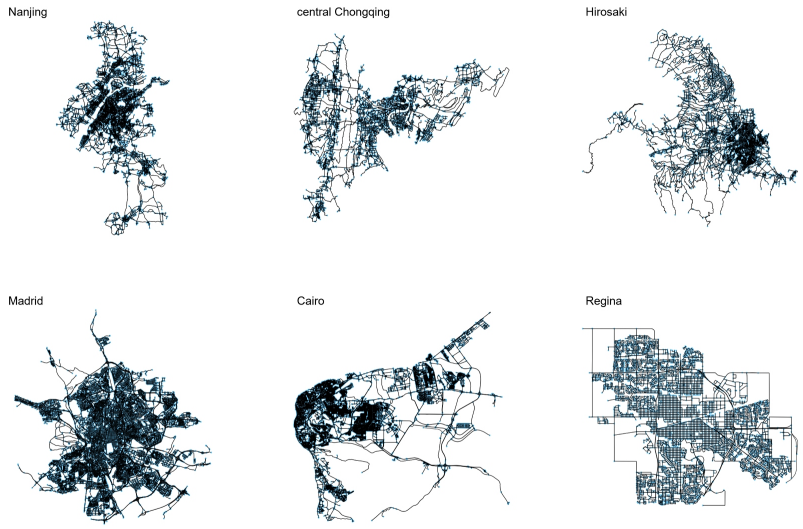

Figure C.10: Random samples from the third of five clusters resulting from clustering in LDA space.

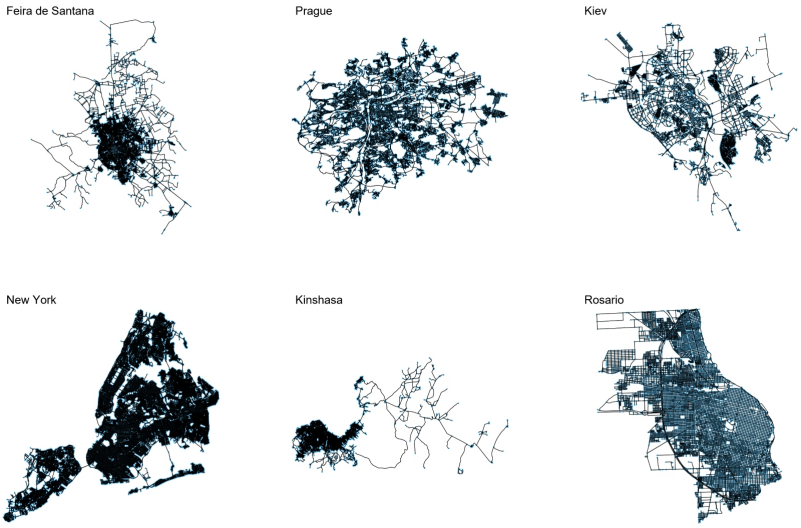

Figure C.11: Random samples from the fourth of five clusters resulting from clustering in LDA space.

Figure C.12: Random samples from the fifth of five clusters resulting from clustering in LDA space.

Bibliography

[1] David D. Woods. Four concepts for resilience and the implications for the future of resilience engineering. *Reliability Engineering and System Safety*, 141:5–9, 9 2015.

[2] Marc Barthelemy. *The Structure and Dynamics of Cities*. Cambridge University Press, Cambridge, 2016.

[3] Luís M A Bettencourt. The origins of scaling in cities. *Science*, 340(6139):1438–1441, 2013.

[4] L. M. A. Bettencourt, J. Lobo, D. Helbing, C. Kuhnert, and G. B. West. Growth, innovation, scaling, and the pace of life in cities. *Proceedings of the National Academy of Sciences*, 104(17):7301–7306, 2007.

[5] Elsa Arcaute, Erez Hatna, Peter Ferguson, Hyejin Youn, Anders Johansson, and Michael Batty. Constructing cities, deconstructing scaling laws. *J. R. Soc. Interface*, 12, 2013.

[6] Jules Depersin and Marc Barthelemy. From global scaling to the dynamics of individual cities. 6(10):2317–2322, 2018.

[7] Xavier Gabaix. Zipf's Law for Cities: An Explanation. *The Quarterly Journal of Economics*, 114(3):739–767, 1999.

[8] E Manley and T Cheng. Exploring the Role of Spatial Cognition in Predicting Urban Traffic Flow through Agent-based Modelling. *Transportation Research Part A: Policy and Practice (2018) (In press).*, 2 2018.

[9] Meisam Akbarzadeh and Ernesto Estrada. Communicability geometry captures traffic flows in cities, 8 2018.

[10] Vahid Moosavi. Urban morphology meets deep learning: Exploring urban forms in one million cities, town and villages across the planet. 2017.

[11] Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 07 2017.

[12] United Nations Statistics Division Demographic Statistics Database. City population by sex, city and city type. http://data.un.org/Data.aspx?d=POP&f=tableCode:240, 1998.

[13] Inc. Environmental Systems REsearch Institute. Esri shapefile technical description. https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf, 1998.

[14] Graph Drawing Steering Committee. The graphml file format. http://graphml.graphdrawing.org/index.html, 2000. Accessed: 27.02.2019.

[15] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (PCA). *Computers and Geosciences*, 1993.

[16] Hervé Herv´, Hervé Abdi, and Lynne J Williams. Principal component analysis. 2010.

[17] CM M. Bishop. *Pattern Recognition and Machine Learning (up to ch.7)*. 2006.

[18] Ludger Rüschendorf. The Wasserstein distance and approximation theorems. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 1985.

[19] L. Kantorovitch. On the Translocation of Masses. *Management Science*, 2008.

[20] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed March 20, 2019].

[21] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[22] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[23] Ulrike von Luxburg. A Tutorial on Spectral Clustering. 11 2007.

[24] R Milo, S Shen-Orr, S Itzkovitz, N Kashtan, D Chklovskii, and U Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 10 2002.

[25] Alexandru Topirceanu, Alexandru Iovanovici, Mihai Udrescu, and Mircea Vladutiu. Social cities: Quality assessment of road infrastructures using a network motif approach. *2014 18th International Conference on System Theory, Control and Computing, ICSTCC 2014*, pages 803–808, 2014.

[26] Nadav Kashtan, Shalev Itzkovitz, Ron Milo, and Uri Alon. Mfinder tool guide. *Department of Molecular Cell Biology and Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot Israel, Tech Rep*, 2002.

[27] Florian Rasche and Sebastian Wernicke. FANMOD: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 02 2006.

[28] Sebastian Wernicke. A faster algorithm for detecting network motifs. In Rita Casadio and Gene Myers, editors, *Algorithms in Bioinformatics*, pages 165–177, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[29] D. Blei, M. Jordan, and A. Y. Ng. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[30] Y. Chang and J. Chien. Latent dirichlet learning for document summarization. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1689–1692, April 2009.

[31] X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In

Seventh IEEE International Conference on Data Mining (ICDM 2007), pages 697–702, Oct 2007.

[32] Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM.

[33] T Benyamini and Y. Teboulle. Fanmod command-line version, 2006. [Online; accessed March 20, 2019].

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

The structural DNA of Cities

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Krummenacher	Franziska

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 8.4.2019	F. Krummenacher

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.