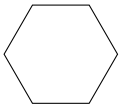
Master Thesis, Institute of Computer Science, Freie Universität Berlin

Biorobotics Lab, Intelligent Systems and Robotics

Temporal Analysis of Honeybee Interaction Networks

based on Spatial Proximity



Alexa Schlegel

Matriculation number: 4292909

alexa.schlegel@fu-berlin.de

Supervisor: Prof. Dr. Tim Landgraf, Freie Universität Berlin

Second Supervisor: Dr. Philipp Hövel, Technische Universität Berlin

Berlin, March 24, 2017

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den March 24, 2017

Alexa Schlegel

Everything we hear is an opinion, not a fact.
Everything we see is a perspective, not the truth.

— Marcus Aurelius

Dedicated to my parents and my brother.

Abstract

TODO

Zusammenfassung

TODO

Contents

Chapter 1

Introduction

A social insect society is formed by thousands of individuals, which continuously move and interact with each other inside a dark nest. Honey bees are organized in colonies, which form a complex and dynamical system. Observing individual honey bees and their interactions with each other is, therefore, vital for understanding collective behavior, decision making, and organization of tasks within the colony.

Within the BeesBook[1] project of the Biorobotics Lab of Freie Universität Berlin Wario et al. [30] developed technologies to automatically track all individuals of a honey bee (Apis mellifera) colony, who are inside the honeycomb.

Shortly after birth, each bee is marked on their thorax by using circular 12-bit tags (figure 1.1) and then added to the observation colony. Four cameras, taking three pictures per second, observe the comb over a period of nine weeks. An image analysis pipeline evaluates each picture automatically. The resulting data set contains, for each point in time, the exact position of each bee on the honeycomb, its decoded id, and its age.

In this thesis, temporal worker-worker interaction networks, based on spatial proximity, are derived from the described data set. Each node in the network is a bee, and a link between two individuals results if they are located close each other. The temporal networks are aggregated for several points in time. Social network analysis methods are applied to determine the usefulness and the characteristics of the resulting networks and its communities.

Until now, social network analysis has been applied to only a subset of a honey bee colonies live, simply because the data were not available to this extent and quality so far.

1.1 Motivation

Most of the studies in the field of animal social network analysis, especially when analyzing the behaviour of social insects colonies, only include a reduced subset of the colonies life, due to a large number of individuals. Usually, the reduction is carried out on three levels (1) time and resolution, (2) space, and (3) number of

[1] `http://beesbook.mi.fu-berlin.de/wordpress/`; Last accessed: 16.03.2016, 12:05 p.m.

Figure 1.1: Tagged bees inside the hive.

individuals.

Labeling only a subset of the colonies individuals, a short observation period, low resolution and manually extracting information from photos or videos is very common in behavioral sciences [20, 27].

Baracchi and Cini [3] observed 300 bees out of a colony with 4000 individuals over a period of ten hours. One of the two sides of the hive was observed by taking a photo each minute. Coloured numbered discs were used for individually marking bees. The analysis of a weighted and undirected worker-worker interaction network revealed a highly compartmentalized structure inside the honey bee colony. Depending on the age, bees occupy different areas of the comb and correspond to different tasks. Also, the contact is limited within groups. Blonder and Dornhaus [5] color painted all individuals of ant colonies (size 6-90 for each colony) and filmed the colonies for 30 minutes. Interactions between individuals were manually extracted by watching the videos. Using time-ordered (dynamic) networks they analyzed the temporal and spatial dynamics of information flow.

Recently, automated tracking of insects has become technically feasible [30, 9, 12]. Using automated high resolution tracking data, which includes all individuals of the complete comb over a long time period allows for more advanced analysis focusing on temporal dynamics. Mersch et al. [18] automatically tracked all individuals of six ant colonies over a period of 41 days. Applying the Infomap community detection algorithm to the physical contact networks for each day, revealed three distinct and robust groups. Each group represents a functional behavioral unit, with individuals changing groups as they age.

The majority of social insect interaction networks studies, due to previously technical limitations, aggregate temporal tracking data into a single static network [17, Chapter 15]. Automatic tracking allows shifting more towards the temporal and dynamic investigation.

1.2 Research Goal

The aim of this thesis is to investigate whether the provided data set of tracked honey bees is useful for creating worker-worker interaction networks using spatial proximity as a proxy for interactions between bees. Thus, I need to implement a pipeline to generate networks out of the given data set. Furthermore, I want to find out if the resulting networks are suitable for social network analysis.

I want to achieve my research goals by answering the following questions:

1. *Is it possible to infer (temporal) networks with the provided honey bee tracking data?*
   What challenges and limitations does the data set imply? What pipeline parameters are necessary?
2. *What kind of worker-worker interaction networks emerge and how are they structured?*
   What is their topology, what properties are characteristic and how do they differ from randomly generated networks?
3. *Does the network display a meaningful community structure?*
   Are those communities robust regarding pipeline parameters?
4. *How are the identified communities characterized?*
   Do they reflect already known colony behavior concerning age and spatial distribution?
5. *How do these communities emerge over time?*
   Are they stable regarding their properties? How do members move between communities?

This work is meant to be the foundation for further more hypothesis-driven research using a network science approach to study and evaluate the complex system of honey bee colonies and their collective behavior.

1.3 Methodology

The methodology of this thesis follows a standard explorative data analysis procedure, mainly to understand the given data set and estimate its quality. The elaborated characteristics of the dataset are then used to define parameters and steps for the network extraction process. This pipeline is tested, reviewed and refined in an iterative process. After each step of refinement, the resulting network is evaluated using the following approaches: (1) check its properties by investigating the effects of pipeline parameters, and features (2) verify its quality using the provided age information of bees, (3) comparing it to a random graph model (Erdös-Renyi), and (4) repeatability of known results and facts concerning its community structures.

1.4 Outline

This thesis is organized as follows. Chapter 2 gives a short introduction into social network analysis (SNA) and defines network measures, terms, and algorithms used throughout this work. In chapter 3, a brief summary of the current state of research concerning social insect networks, temporal networks and community detection in animal social networks is given. Chapter 4 describes my research approach in general and how the pipeline produces networks out of the given dataset, what steps are needed and what parameters it uses. Also, I explain and justify what decisions I took during the network analyses and community detection process. Chapter 5 reports the results of the network analysis and the characteristics of the extracted communities. Finally, in chapter 6 I explore the results, discuss limitations and conclude with directions for future work.

Chapter 2

Theoretical Background

The following chapter gives a short introduction into social network analysis (SNA). It introduces animal interaction networks as a special type of network. It defines terms and concepts used throughout this work and explains networks metrics and algorithms of which we will make use of.

A *social network* is a representation of a social structure comprising actors such as individuals, affiliations, as well as their social interactions. The network model conceptualizes social, economic, or political structures as lasting patterns of interactions between actors [31]. In mathematical terms, networks are graphs, and thus consist of *nodes* (vertex, representing individuals), and *links* (edges, relationships or interactions). Social network analysis provides a set of methods, measures and theories, borrowed from network and graph theory, to investigate social structures and its dynamics.

This work is focusing on the special case of animal social networks. Networks where individuals are nodes and edges are defined as interaction events between individuals are called *interaction networks*, sometimes also contact networks. According to Charbonneau et al. [8] those interactions used as an edge can be of four different types when looking at animal networks: spatial proximity, physical contact (usually with antennae, "antennation"), a food exchange event (trophallaxis), or specific communication signals.

Edges can be directed (e.g. trophallaxis) or undirected, weighted or unweighted. The edge weights represent the strength of the relationship commonly the number or duration of interactions is used [11].

2.1 Network Measure and Metrics

The following definitions are mainly taken from Barabási [2] and Newman [21]. the gray box summarizes the basic variables and terms of this work. [TODO: Box referencing as table and align bottom.]

Network size N is the total number of nodes, respectivly animals in a network.

Number of links L is the total number of links, social interactions, in a network.

Edge weight w_i of an edge l_i is an indicator of how important that edge is.

Component is a subnet of nodes in a network, so that there is a path between any two nodes that belong to the component.

Degree k_i of a node n_i represents the number of edges a node has; so the number of other animals this animal interacts with.

Average Degree $\langle k \rangle$, the number of animals one animal interacts with on average.

Path length d the shortest number of links between two nodes.

Average path length $\langle d \rangle$ is the average of all shortest path between all pairs of nodes.

Diameter $d_{\texttt{max}}$ is the longest of all path length. The distance between the two furthest nodes, the longest possible path length in the network.

Density D is the number of realized links divided by the number of theoretically possible links is defined as

$$D = \frac{2L}{N(N-1)}$$

Is it independent from the edge weights.

Strength s_i of a node is also called the weighted degree. It measures the total weight of edges connected to a node n_i and is definded as

$$s_i = \sum_{j=1}^{n} w_{ij}$$

according to Barrat et al. [4]. The average strength denoted as $\langle s \rangle$.

Global clustering coefficient C_Δ also called transitivity. According to Wasserman and Faust [31] it is defined as

$$C_\Delta = \frac{3 \times \texttt{number of triangles}}{\texttt{number of connected triples}}$$

.

Local clustering coefficient c_i of a node n_i quantifies how close its neighbours are to being a clique (complete graph).

Centrality When looking at the networks local structure (node level), it is possible to identify nodes, which are important or central to the network, regarding different aspects. This concept is called *centrality* and measures the influence of a node in a network. [21] In the course of analysing networks and their local node level structures, you will find and encounter the most important (central) nodes and vertices by indicators of centrality. These indicators give values to the nodes and therefore they can be listed in a way of importance.

The the weighted versions of betweenness and closeness using Dijktra and the inverse of the edge weights.

Degree Centrality Degree centrality C_D^i of a node n_i is the normalized degree k_i in relation to the whole network, it is calculated as follows:

$$C_D^i = \frac{k_i}{N-1}$$

Eigenvector Centrality The eigenvector centrality x_i of a node n_i is the sum of its connections to other nodes, weighted by their centrality.

$$x_i = \frac{1}{\lambda} \sum_j A_{ij} x_j$$

It is like a recursive version of degree centrality. So a nodes importance (centrality) is inceased by having neighbours that are themselves important. Eigenvector centrality gives each vertex a score proportional to the sum of the scores of its neghbours. [21]

Closeness Centrality Is is the average length of the shortes path between node n_i to all other nodes in the network. The more central a node is the closer it is to all other nodes. Mean distance from a node to other nodes. [21]

$$C_C^i = \frac{N}{\sum_j d_{ij}}$$

Betweenness Centrality It measures the extend to which a node lies on paths between other nodes. Nodes that occur on many shortest paths between other nodes have higher betweenness that those that do not.

2.2 Temporal Networks

When modeling temporal or so-called dynamic networks two main approaches exists (1) time-aggregated (discrete), where the data is aggregated either in a disjoint, overlapping or cumulative snapshot, and (2) the time-ordered (continuous) approach, with interactions having a start and end timestamp [19, 25, 6].

The time-aggregated approach aggregates the data for each snapshot and therefore reduces the available information per edge. In contrast, the time-ordered approach keeps the information for each edge, when the interaction occurred and how long it lasted. It provides a detailed insight when timing and order of interactions are important. And therefore it can be used to model the topological flow information through a network.

Choosing suitable time intervals for aggregating is challenging [25], but a lot of methods for analyzing those networks already exists, whereas for time-ordered networks, only limit toolset is available. In time-aggregated networks, the modeling nodes and edge weights can be challenging when taking into account that interactions, which took place earlier or later in time are weighted accordingly.

2.3 Community Detection

To understand the large-scale structure of networks, one can look at the network's community structure. Communities are naturally occurring groups within a network, usually also called clusters, cohesive groups or modules and have no widely accepted, unique definition [24]. For my work, I adapt the definition according to Barabási [2]: "In network science, we call a community a group of nodes that have a higher likelihood of connecting to each other than to nodes from other communities." [2, p. TODO]. In contrast to a simple graph partition, the number and size of communities is not predetermined or set in advance.

Communities in animal social networks refer to groups of individuals that are associated more with each other than they are with the rest of the population. These communities reflect an intermediate level of social organization, which is located between the individual and population level [10].

There are a lot of different approaches and algorithms who adress the detection of communities. Fortunato [13] gives an extended overview of the various types of community detection algorithms. Explaining any of those would be beyond the scope of this work. For example, traditional methods include algorithms based on graph partitioning, hierarchical clustering, and spectral clustering. There are also divisive and agglomerative algorithms. The algorithms used in this work are described in the following sections and include the leading eigenvector [22] and walktrap [26] algorithm.

2.3.1 Modularity

Modularity is a quantity, that measures the quality of a partitioning. It can be used to compare a community partition to another and decide for the better one. Modularity optimization is also used for community detection algorithms.

A high modularity of a network indicates more connection between nodes within a community and fewer connections between nodes of different communities. The

basic idea is: If the fraction of links inside the community is higher, than expected in the same community of a related random graph having the same degree distribution, then it is a community in the sense of modularity. This difference is summed up and normalized. If all nodes fall into one community the modularity is 0. Fewer links inside the community than expected result in a negative value, otherwise positive.

2.3.2 Leading Eigenvector and Walktrap

The *leading eigenvecor* algorithm was proposed by Newman [22]. It uses the eigenvectors of matrices for finding community structures in networks. It is a top-down hierarchical approach that optimizes modularity. The algorithm starts with all nodes inside one community, therefor a modularity of 0. In each step, the network is split into two parts, so that the modularity of the new separation increases. The splitting is done by first calculating the leading eigenvector of the modularity matrix and then splitting the graph in a way that modularity improvement is maximised based on the leading eigenvector. The algorithms stops if the modularity is not increasing anymore.

This *walktrap* algorithm by Pons and Latapy [26] is based on random walks. The authors consider random walks as a tool to calculate similarity between nodes of a network. It uses a bottom-up hierarchical approch, that means the algorithms start with each node it is own community. The basic idea of walktrap is, that short distance random walks (the step size is a parameter) tend to stay in the same community, because there are only a few links that lead outside a given community. The results of these random walks are used to merge separate communities. Again modularity can be used to cut the dendrogram in an optimal place.

2.4 Communities in Temporal Networks

According to Aynaud et al. [1] and Bródka et al. [7] there are three main approaches for community detection in temporal networks (also called community tracking): (1) using a static community detection algorithm on several snapshots and then solving a matchig problem, (2) using algorithms who are directly suited for temporal networks and (3) using incremental or online algorithms when processing data streams. For each of the three approaches, several mehods already exist.

As community tracking is not the main focus of this work, I chose to apply the most intuitive approach aout of approach (1): detecting static communities for each snapshot and then matching those communities using set theory. Two communities at successive timesteps are matched if they share enough nodes. The *match value* (between 1 and 0) between two communities C and D according to [15] is defined as:

$$\texttt{match}(C, D) = \texttt{min}\left(\frac{|C \cap D|}{|C|}, \frac{|C \cap D|}{|D|}\right) \tag{2.1}$$

A high match value accurse, when two communities share a lot of nodes and are of a similar size. Communities with the highest value are matched. A threshold should be applied to more precicley define what “share enough nodes” means.

Chapter 3

Related Work

Look at [27] for static network analysis stuff they measured (ants, only small amount with observed interaction using 20 Minute videos).

3.1 Networks in social Insects and Honey Bees

Structure of the social network and its influence on transmission dynamics in a honeybee colony

In *Social Insects: A Model System for Network Dynamics* [8] a good overview is given about Role and Types of Networks in Social Insects.

3.2 Spatial Proximity Networks

TODO short summary, also on network sience methods:
Long-term dynamics in proximity networks in ants [16]

Contact networks and transmission of an intestinal pathogen in bumble bee (Bombus impatiens) colonies [23]

3.3 Temporal Aspects in Networks

3.4 Network Analysis Tools

Chapter 4

Approach and Implementation

This chapter describes the workflow and implementation I applied to reach my research goals. The preparation and processing of the data are primarily driven by a combination of exploratory and iterative processes.

I primarily follow Farine's and Whitehead's [11] steps and key considerations for social network analysis to non-human animal data. [TODO: steps kurz zusammenfassen, und unterschiede zu mir erklären] Figure 4.1 visualizes the adapted and resulting process. [TODO: Schritte im Bild nummerieren und im Text verwenden]

In the first step, the data set was analyzed to form a general understanding of the given dataset, its structure, quality, and characteristics.

The results of this investigation are used to define nodes and infer associations to build the network, respectively derive the parameters for the network pipeline.

The resulting networks are analyzed by using network science tools and methods. [hier unbedingt methoden aufzählen]

[TODO: umfomulieren, Begruendun Auswahl der Attribute, und welche Test habe ich genau gemacht.] For testing underlying hypothesis the networks are attributed with the bees spatial information and their age.

4.1 The Dataset

The dataset derives from high resoluted video files, that capture tagged honey bees of one colony in an one-frame observation hive. The individuals of the colony, including about 3200 bees over a nine weeks period, were tagged with circular 12-bit markers (figure 1.1, section 1). Two cameras per side filmed the complete honeycomb permanently. Figure 4.2a illustrates the camera setup. The *recording period* lasted nine weeks (63 days), from 19.07.2016 until 19.09.2016, with some interruptions due to maintenance work and technical failures. [TODO: verlinken Appendix, uebersicht ueber gute und schlechte Zeiträume][TODO: Abbildung Aufnahmezeitraum]

All four cameras (4000×3000 pixels) record 3.5 frames per second. An image analysis pipeline [30] detects all bees in each frame. The resulting detection data is stored in a binary file format. An python library[1] provides an frame-level access to those

[1]The library is called `bb-binary` and is created by the Biorobotics Lab. It can be found on github:

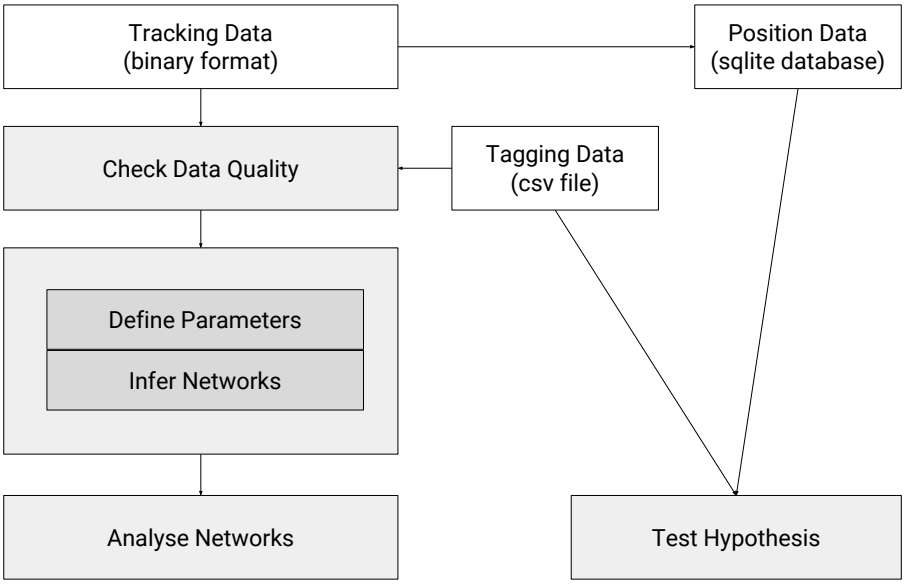


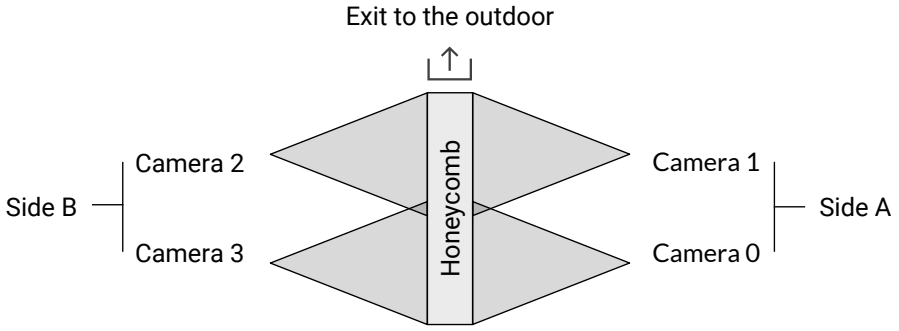
Figure 4.1: Steps of the research approach [TODO: anpassen, iterationen, hypothesis raus, evtl. so viele Schritte im Bild wie Abschitte im Text, zumindest muss das irgendwie anders.]

binary files. The size of the dataset is 470 GB, about 7.5 GB of binary data per day.

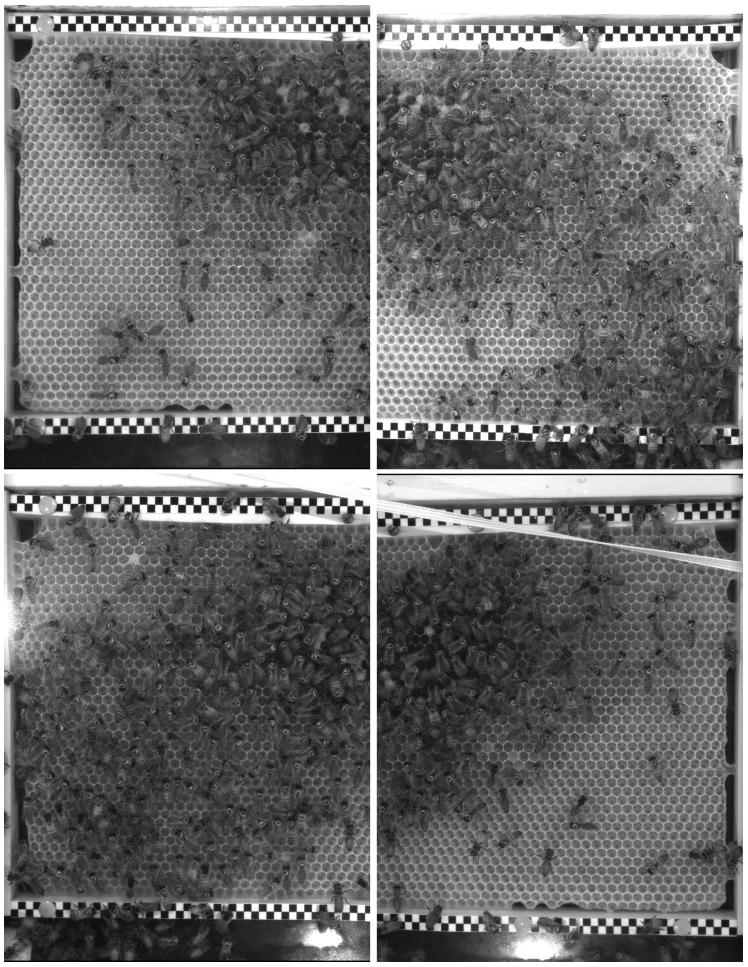
The tagging period (67 days long) started on 28.06.2016 and lasted until 02.09.2016. Exactly 3.191 bees were tagged. The young bees, which are raised in a separate hive, were tagged and then added to the observation hive, about noon each day. Figure 4.3 shows the frequency of the tagging. The hatching day for each bee is documented; therefore the age of each bee at a particular point in time can be calculated.

[TODO: Begruendung des gewaehlten Zeitraumes: Funktionstüchtigkeit und Altersverteilung] For further analysis, I chose three days (20.08., 22.08., and 24.08.) highlighted in figure 4.4.

`https://github.com/BioroboticsLab/bb_binary`; Last accesed: 2106-02-16, 04:28PM



(a) Camera setup Each side of the honeycomb is filmed by two cameras. The two cameras per side do overlap, bees inside this are are detected from both cameras.



(b) Images for each camera Top: Side A, bottom: side B

Figure 4.2: Observation setup

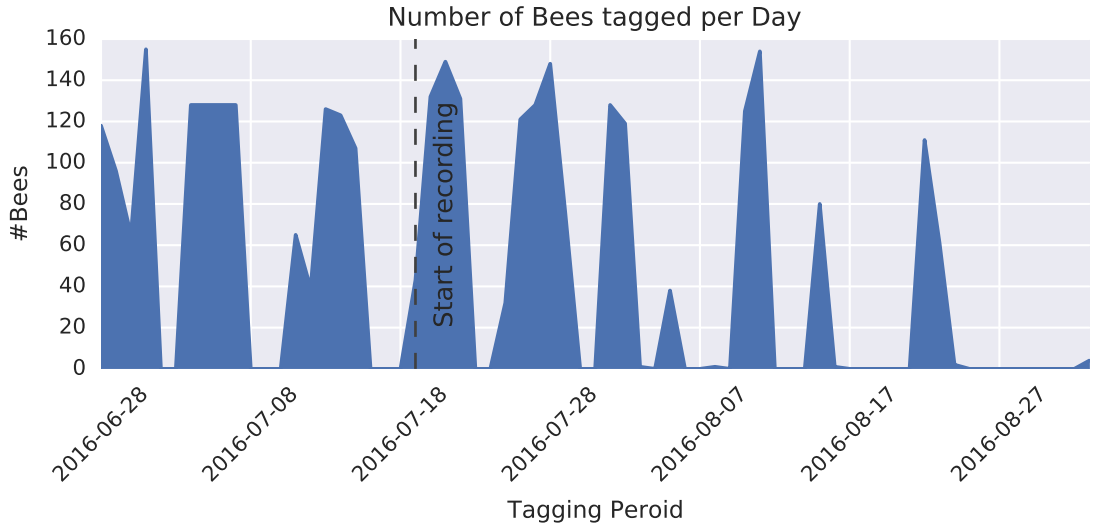


Figure 4.3: Tagging frequency: The bees were primarily tagged during the week. On average 48 bees were tagged each day, considering only tagging days, the average is about 91 (±50) bees (median 118). [TODO: combine with other image or make nicer!]

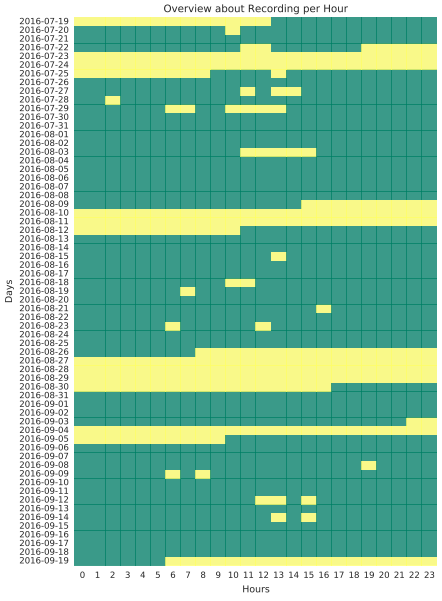


Figure 4.4: Recording season with maintainance and failures *Green* indicates recording went without any big interruption; *Yellow* indicates maintainance work or technical failures of one or all cameras. This is calculated using the expected number of files produced by each camera per hour. [TODO, reduzieren auf eine Info pro Tag (keine stuendliche aufloesung), kombinieren mit anzahl der getaggten bienen pro tag, und welchen Zeitraum hab ich nun verwendet], ausserdem Zeit von links nach rechts!, evtl. kein Datum, sonder Tage durchnummerieren

4.1.1 Data Scheme

The data is organized in so-called *frame containers*. Each frame container corresponds to one video file of a single camera and consists of about 1024 *frames*. So the frame container specifies the camera (*camId*), which took the video. Each frame holds a list of bees, which were detected by the image analysis pipeline and is attributed with a *timestamp*.

A bee *detection* has, among others, the following attributes:

xpos:	x coordinate of bee with respect to the image in pixel
ypos:	y coordinate of bee with respect to the image in pixel
decoded ID:	decoded 12-bit ID
cam ID:	ID of the camera $0, 1, 2, 3$
timestamp:	unix timestamp

The data can be accessed iterating on frame level, using a start and end timestamp, for specifying a time interval. The complete data scheme can be found on GitHub[2].

4.1.2 ID Probabilities, Confidence Level, and Quality

Twelve bits can encode the identity of 4096 bees. Each bit of the decoded ID is not a one or zero, but represents a probability between 0 and 255, normalized to a value between 0 and 1. Therefore, a bit indicates the reliability of the image analysis pipeline. I define the confidence c for a bit b, analougus to Leon Sixt [29, p. 14], as $c(b) = 2 \cdot |b - 0.5|$. The confidence of a decoded ID is, accordingly, the minimum of all twelve bit confidences.

The amount of data that remains for further processing and its correctness is highly dependent on the chosen level of confidence.

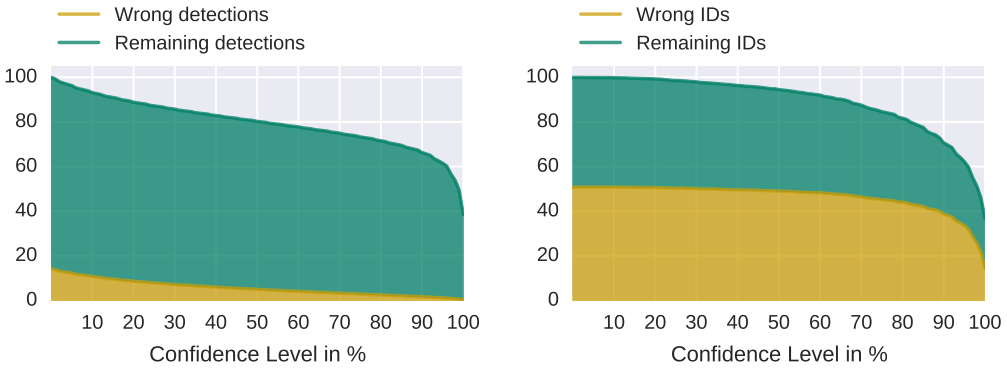
Additionally, I use the age information of the bees to check the quality of the remaining data. I examined (1) the number of detections and (2) the number of unique IDs, depending on the chosen confidence. For each detection, the age was calculated. A detection with a negative age is counted as a wrong detection. I assumed that the number of wrong detections also occurred among detections with a positive age, but remained unseen; therefore I doubled the error, to be more accurate[3].

For (2), analogous a unique ID with a negative age is counted as a *wrong* ID.

As expected, with an increasing confidence, the remaining data and unique IDs do decrease (figure 4.5). Also even though the number of wrong detections decreases

[2] `https://github.com/BioroboticsLab/bb_binary/blob/master/bb_binary/bb_binary_schema.capnp`; Last accessed: 2106-02-16, 04:46PM

[3] On 26.07.2016, about half of the bee tags (2014 out of 4096) were assigned. Because of this, that day was chosen to determine the effects of the level of confidence on data quality and the amount of remaining data.



(a) Detections **(b) IDs**

Figure 4.5: Quality of detection and IDs *Green* represents the number of remaining detections (from all, confidence 0%) and remaining IDs (from 4096 possible IDs). *Yellow* indicates the fraction of wrong IDs and detections in relation the remaining IDs and detections. (10 minute dataset, 26.07.2016, 4 p. m., all cameras)

steadily with an increasing confidence level, the number of wrong IDs only starts to decrease with a very high level of confidence.

With a confidence level of 100%, 30.2% of the remaining unique IDs are wrong (have a negative age), corresponding to only 2.5% wrong detections of the remaining detections. Therefore, to obtain a more reliable dataset, wrong detections need to be filtered out, independently of the confidence value.

4.1.3 Time Series of Bees and Bee Pairs

The dataset, is transformed to binary *bee time series*, depicted in figure 4.6 (left and middle). A time series of a bee is a sequence of zeros and ones indicating the

Frame container	A container for all frames, which belong to a specific video file of a certain camera.
Frame	This includes all detections of one camera image, at a certain point in time.
Detection	A detection of a bee at a certain point in time.
Decoded ID	Identifyer of a bee consisting of 12 probability values, representing 12 bits.
Confidence	Value between 0% and 100%).
ID	The decimal representation of an decoded ID, after applying a certain confidence value.
Bee time series	Binary sequence, indicating the absence and presence of a certain bee in a particular time interval.
Pair time series	Binary sequence, indicating the absence and presence of two bees in a particular time interval.

absence and presence of a bee over a specified time interval. I examined the effect the level of confidence has on the bee time serie. As expected, with an increasing confidence level the average gap length decreases and the overall number of gaps increases (4.7).

The number of gaps those bee time series have, is important, because in a later step I want to extract pairs of close bees, who are present at the very same time. I call those *pair time series*, as shown in figure 4.6 (right). So a lot of gaps in bee time series, could lead to a lot of gaps in the pair time series.

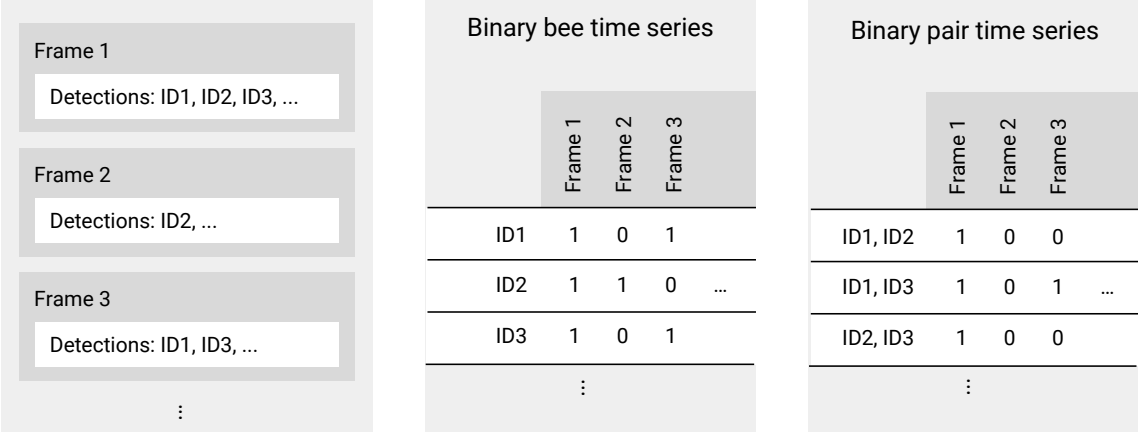
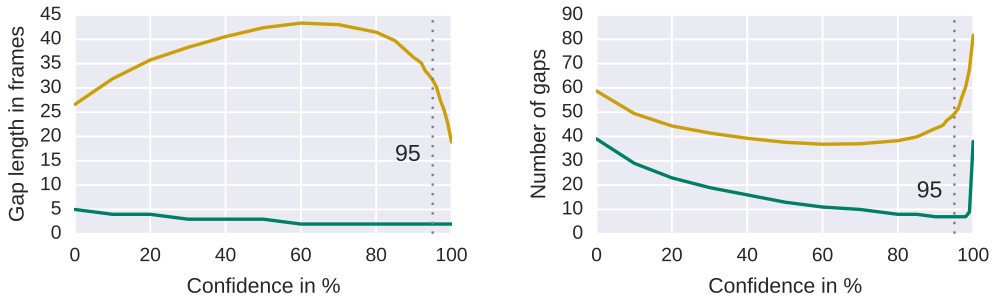


Figure 4.6: Structure of dataset *Left*: original dataset - containing a sequence of frames with bee detections; *Middle:* binary bee time series - zero and one indicating absence and presence of the a bee; *Right:* binary pairs time series - zero and one indicating the absence and presence of two bees in the same frame.



(a) Length of gaps **(b) Number of gaps**

Figure 4.7: Influence of the confidence level of the length of gaps and number of gaps: With an increasing level of confidence the average gap length decreases and the number of gaps per bee series increases. (Dataset: 26.07.2016, 16:00-16:05)

4.1.4 Detection Frequency Filter

A good indicator, weather a bee is alive and present on a specific day, is the detection frequency of IDs. The hypothesis is: Bees with IDs who have a low detection rate

are not physically present in the hive (do not exists at that day), due to detection errors of the image analysis pipeline. To check this hypothesis, I investigate weather there is a correlation between age of the bee and detection frequency.

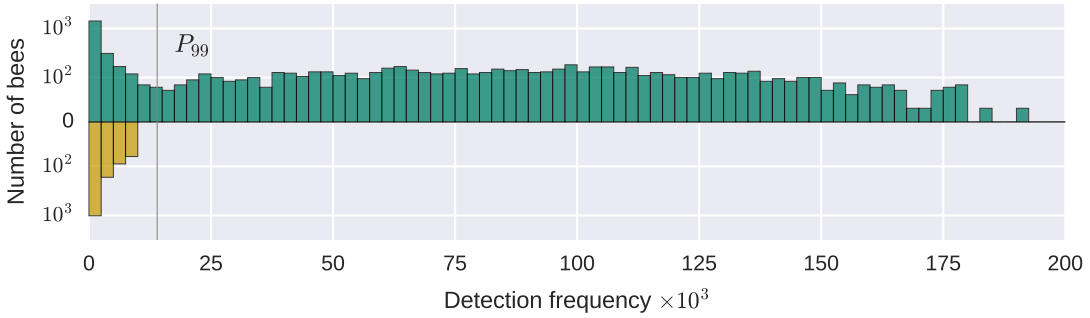


Figure 4.8: XXX XXX

IDs with a negative age are on average less detected than IDs with a positive age.

IDs who definately exists, but their age can to be determinded are excluded from the analysis completely. These are bees, who were tagged later ($n = 10$)[4] and IDs whos detection frequency is absurd high but there age is unknown ($n = 7$)[5]

For each analysis day the number of detections per ID is obtained, excluding the mentioned IDs above. The frequency distribution tells that, IDs with a negative age are detected less often (704 frames ±65) than bees with a positive age (36.603 frames ±2.345). The cutoff is 99% of the negative IDs distribution. All IDs with a detection frequency below 4737 ± 644 frames are discarded. A list with possible (valid) IDs is kept for each day. Using this list, faulty detections can be filtered out beforehand.

[4]id= [2, 74, 2045, 3172, 3764, 3796, 3827, 3836, 3844, 3940]
[5]id=[has changed! 17, 168, 801, 888, 2045, 2357, 2607]

4.1.5 Implications

[TODO] For further analysis I use the following days: 14.08.2016, 17.08.2016, 20.08.2016, 02.09.2016 with a confidence level of 95%. This period is chose because of XY.

Because bee time series contain a lot of short gaps (mean = 3, 95% confidence), the inferring of edges (bees who are close to each other at the same time), should be not that strict, or at least variable. This has to be taken into acoount, when looking at spatialy close bees.

4.2 Inferring Networks

The following part describes the pipeline for generating spatial proximity networks out of honey bee tracking data. A node in the network is a bee. They are distinguished by IDs. Only bees are in the network who interact at least once with another bee.

undirected and weighted, aggregated networks

Two bees are associated (spatially close to each other), if their distance is minor to a *maximum distance*. As everything is very close in a bee hive this value is hard to choose. Only this criteria is very week, meaning having a resolution of three frames per seconds results in interactions which could only last for 0.33 seconds. So an additional parameter the *minimum contact duration* is introduced, it is the minimum time they have to spend at least nearby to be called associated.

Taking the fragmentation of tracks into account, it is obvious that two bees could be nearby but not at the very same time, but slightly shifted. So the minimum contact duration would be too errow prone. To overcome this issue one could correct the bee tracks, by filling gaps of varius sizes and interpolating the position of that bee accordingly. This is rather time consuming for this amount of tracking data (TODO: naja so doll auch nicht, scipy.ndimage.morphology.binary_dilation) and also considering, that the tracking data is going to be improved in the future, then manipulating the raw data seems senseless. I rather perform a gap filling (maybe similar to binary dilation) on the time series of pairs, but not on the bee tracks, because this is independent of the input data.

Edges are attributed with two parameters. The first one is the frequency of contacts, so how often they share a close position. The second parameter is the total duration of contact, how many time frames in total they spend close by.

4.2.1 Network Pipeline

The network pipeline takes as input a path to the bb-binary data and outputs a graph in graphML file format. The pipeline takes the following parameters:

- path to data
- confidence in percent
- gap size in frames - this is used to corret the time series of bee pairs
- maximum distance in px - define what close means (spatial proximity)
- minimum contact duration in frames - how many frames bees need to spend nearby
- cutoff in percent - IDs with a number of total detections below X percent of the mean frequency are discarded

- start timestamp - start of network slice
- window size in minutes - size of time window for aggregating the network
- number of used CPUs for parallelization
- year - calculate IDs and set camera setup for 2015 or 2016

The pipeline is parallelized on frame level, that means, each process gets a portion (frames for a timeinterval of five minutes) of the data and extracts interactions/edges. The main process adds everything up and creates a network. The steps are the following:

1. **Filter detections by confidence**
   For each of the four camera the detections are filtered by the confidence level.
2. **Simple stitching**
   Each side of the hive consists of two cameras. The x-coordinates of each detection (of the right cameras) is moved further to the right, also adding an offset of $2 \times$ `maximum distance`. So the left and the right detection of each side of the hive are move into one reference system.
3. **Syncronize Cameras**
   For each side of the hive the cameras need to be syncronized. In the normal case the difference between consecutive frames should be about 0.332 seconds, due to technical problem this value can be lower (0.003) and higher (2.932) at certain times. Cameras 3 and 2 and cameras 1 and 0 are matched, frames without a match are dropped (shorter number of frames, matchen, threshold 0.33/2, minimum).
4. **Discard Detections with certain IDs**
   All detections whos ID is in a list are keept, other detections are discarded. (see frequency filter)
5. **Extract close pairs**
   For each side of the hive, all close pairs according to the maximum distance parameter are calculated and then joined together.
6. **Generate time series of bee pairs**
   The data structure (frames and detection) is transformed to time series of bee pairs.
7. **Correct pair time series.**
   The time series of bees are corrected by filling in the gaps of length `gap size`.
8. **Extract edges**
   The edges and its attributes (frequency and duration) are extracted from the time series of bees using the minimum contact duration parameter. A sequence of at least X ones counts as one interaction. The frequency of those series adn the total duration (number of ones) are the attributes.

4.2.2 Pipeline Parameters

For performing the network analysis, I chose the pipeline parameters as follows:

Confidence As explained in section4.1.2, the confidence is set to 95%.

Maximum Distance I chose the length of a bee body, according to Baracchi and Cini [3], as the maximum distance between two bees (figure 4.9a). The average bee length of 212px (±16px) was determinded by manually measuring the length of all bees ($n = 337$) in four images (one for each camera, 21.07.2016, 03:00PM) using the tool ImageJ[6].

Gap Size The gap size is set to two frames. This value corresponds to the median gap length in the time series of pairs (`mode` = 1, `mean` = 27). [TODO: what dataset was used (95% confidence, XXX% cutOff, XXXpx maximal distance, date, camera)]

Minimum Contact Duration This is set to three frames (one second). This corresponds to Mersch et al. [18], they as well exclude interactions below one second. Looking at the frequency distribution of chains of ones (1, 11, 111, and so on) of the pair time series (after filling the gaps), then: `mode` = 1, `median` = 2 and `mean` = 4. Three frames corresponds to 57% of all chains, this seem to be reasonable. [TODO: what dataset was used (95% confidence, XXX% cutOff, XXXpx maximal distance, date, camera)]

The networks are not thresholded (according to [11]).

4.3 Static and Temporal Analysis

Despite the possibility of generating networks of different granularity (resolution is minutes), here for further analysis daily networks (10h, two hours after sunrise until zwo hours before sunset) are aggregated.

Wey et al. [32]

4.3.1 Static Network Analysis

The following network properties were analysed for a static day and hour network. TODO: list of properties. (similar to what others have done) nodes, edges, density, diameter

[6] `http://imagej.net/Welcome`; Last accessed: 22.02.2016

4.3.2 Temporal Analysis

three day networks (2 days gap)
one network 2 weeks later

4.3.3 Community Detection

I tested all community detection algorithms implemented in python, to find an algorithm, which works well for my case of animal social networks. The three most common python libraries for network analysis were reviewed: NetworkX[7], igraph[8], and graph-tool[9])

The algorithm needs to fulfill the following criteria:

- Support for large and very dense networks ($N > 1000$, $D > 50$ %)
- Support weighted edges
- Fast runtime

Table 4.1 gives an overview about the twelve algorithms reviewed. Five algorithms did not terminate after 15 minutes and were therefore excluded from further investigations. Infomap and label propagation tend to partition all nodes into a single community, this is known especially in dense graphs [33, 13]. The Louvain algorithm is the same as multilevel, but takes longer producing almost the same communities and therefore was also excluded. Walktrap was tested for different step size parameters, as suggested in [26], the communities remained almost the same (only a few nodes switched communities).

I had a closer look at fastgreedy, leading eigenvector, multilevel, and walktrap regarding the number of detected communities and community size for all three networks. Table 4.2 shows the results. All algorithms found at least two communities. Except for leading eigenvector, there is a tendency that a third community exists. I decided to use two algorithms for community detection: leading eigenvector and walktrap. Farine and Whitehead [11] explains that leading eigenvector is often used with animal social networks and works well. Walktrap is chosen for also examining the possible third community.

There are comparative analysis of community detection algorithms, e.g. [33, 14]. They seem to be promising, but assume eighter a power law degree distribution or evaluate networks with a low density, which is not applicable here.

[7] `https://networkx.github.io/`; Last accessed: 16.03.2016, 6:36 p.m.
[8] `http://igraph.org/python/`; Last accessed: 16.03.2016, 6:38 p.m.
[9] `https://graph-tool.skewed.de/`; Last accessed: 16:03.2016, 6:39 p.m.

Table 4.1: Comparing community detection algorithms Comparison of algorithms implemented in python. Criterias are the support of weighted edges, runtime and number of communities. A runtime indicated by − mean no termination after 15 minutes.

	fastgreedy[1]	leading eigenvector[1]	louvain[2]	multilevel[1]	walktrap[1]	infomap[1]	label propagation[1]	edge betweenness[1]	k-clique communities[2]	optimal modularity[1]	spinglass[1]	statistical inference[3]
Edge weights	×	×	×	×	×	×	×		×	×	×	
Runtime in sec	3.6	6.3	11.7	0.7	19.4	13.2	0.2	–	–	–	–	–
Communities	3	2	2	3	2	1	1	–	–	–	–	–
	473 434 15	488 434	469 453	462 427 33	490 431 (1)	922	922					

[1] igraph, [2] NetworkX, [3] graph-tool

Table 4.2: X X

	fastgreedy	leading eigenvector	multilevel	walktrap
Network 1	473 434 15	488 434	462 427 33	490 431 (1)
Network 2	504 467 7	503 475	481 439 58	372 311 294 (1)
Network 3	534 388	537 385	505 415 (2)	310 390 231

4.4 Attributed Data and Hypothesis Testing

Hypothesis
(1) Communities reflect groups of bees working in different areas of the hive and
(2) Communities reflect different age groups

The data which was used to test the hypothesis (1) is saved in a sqlite database for faster access, because using bb_binary (parsing the data over and over again) was to slow. For testing if lists of positions (spatial ditribution) are different the test XY was used [TODO: what to use here]

For hypothesis (2) the data is stored as a csv file of birth dates of each bee. For testing if age goups are different the Kolmogorov Smirnov Test was used.

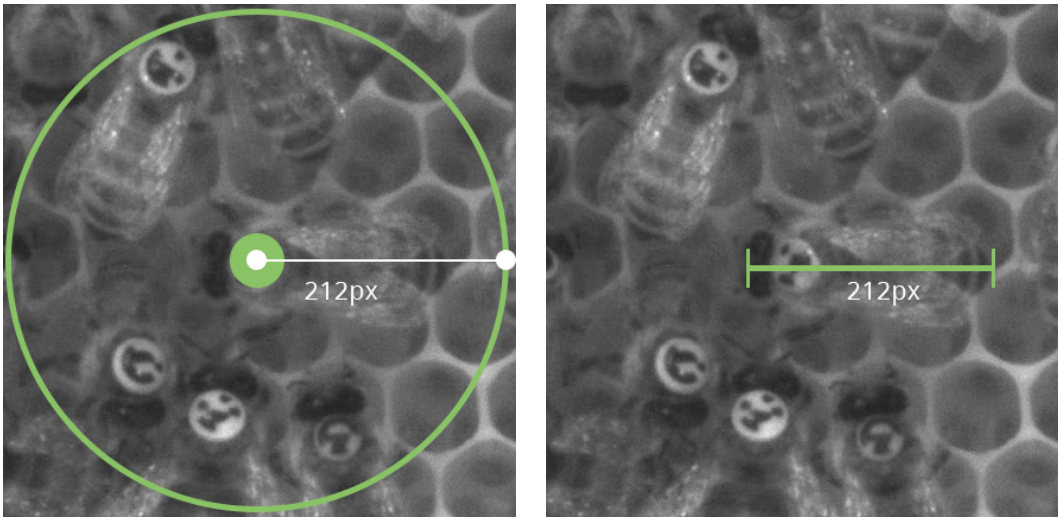
4.5 Implementation, Runtime and Complexity

For implementing the network pipeline python, with pandas and numpy, are used, because the bb_binary library, for accessing the tracking, data is only available in python. The networks, in graphML format, are created using the python library *NetworkX*[10] in version 1.11. iGraph for community detection
some bash scrips for generating multiple networks

bottleneck is reading bb_binary data into pandas dataframes
using multithreading for distribution on frame level (a process gets X frames for processing)

maybe some table with how long nees what with how many cores (hom much RAM and so on)

[10] `https://networkx.github.io/` ; Last accessed: 2017-02-17, 08:07PM



(a) Contact Radius **(b)** Bee Length

Figure 4.9: Distance Between Bees: A length of a bee is chosen as the maximal distance between bees.

Chapter 5

Results

[TODO, highlight more the temporal aspects of this work]

I analyzed three daily networks, these are referred to below as network 1 ($N = 922$), network 2 ($N = 978$) and network 3 ($N = 922$), or g_1, g_2, and g_3. Each network is aggregated for ten hours (108 000 frames) starting at 8 a.m. and lasting until 6 p.m. I chose the 20.08.2016, 22.08.2016, and 24.08.2016, because at this point in time the hive also contained tagged foragers, thus older bees. At the same time young bees were added to the colony, see table 5.1 for details. The age distribution for each network is shown in figure 5.1. The match value, according to equation (2.1) between network 1 and 2 is 85% (833 bees), between network 2 and 3, it is 84% (823 bees) and between network 1 and 3, it is 78% (716 bees).

5.1 Network Topology and Centrality

Each network consists of one giant component. Table 5.2 summarizes basic network properties. The network density id for all networks over 50%. The diameter is for all three networks 3 and the average path length is below two. Compared to an Erdös-Renyi random graph the global clustering coefficient is higher, averaged over 100 runs per network with an SD below 0.00005.

On average a bee is connected to 68%, 54% and 61% percent of all bees in the network. Figure 5.2a show a slighty bimodal degree distribution. The degree distribution does not follow a pow law, which is typical for real world social networks.

The distribution of strength is shown in figure 5.2b and the distribution of edge

Table 5.1: Sampling period Overview of the chosen day networks including the number of added bees and the time they were added to the hive.

	20.08.16	21.08.16	22.08.16	23.08.16	24.08.16
Network ID	1	-	2	-	3
Number of added bees	0	0	110	60	0
Time added	-	-	2 p.m.	6 p.m.	-

Table 5.2: Global network properties N is the number of nodes, L the number of edges, D is the diameter, $\langle d_{\texttt{max}} \rangle$ is the average path length, C_Δ the global clustering coefficient, $C_\Delta^{\texttt{rand}}$ is the global clustering coefficient for randomized graph, $\langle k \rangle$ the average degree and $\langle s \rangle$ represents the average strength, as introduced in section 2.1.

	N	L	D	$\langle d_{\texttt{max}} \rangle$	$\langle d \rangle$	C_Δ	$C_\Delta^{\texttt{rand}}$	$\langle k \rangle$	$\langle s \rangle$
Network 1	922	291179	0.69	3	1.32	0.79	0.68	631.62	5680.17
Network 2	978	256066	0.54	3	1.46	0.72	0.54	523.65	3977.94
Network 3	922	259421	0.61	3	1.39	0.75	0.61	562.74	4205.99

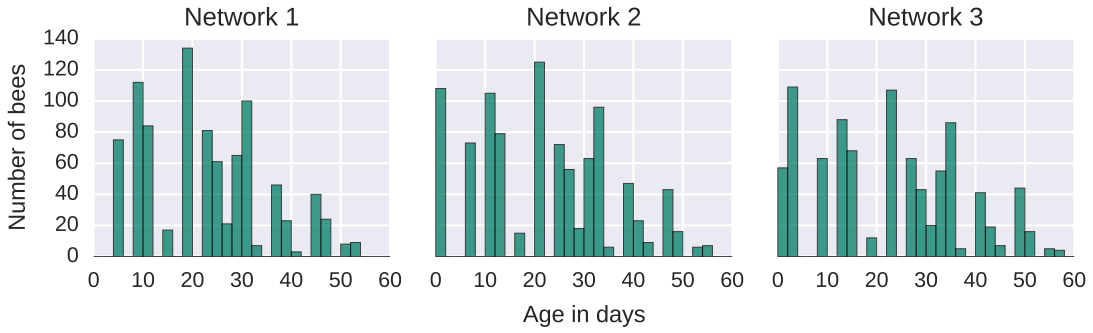
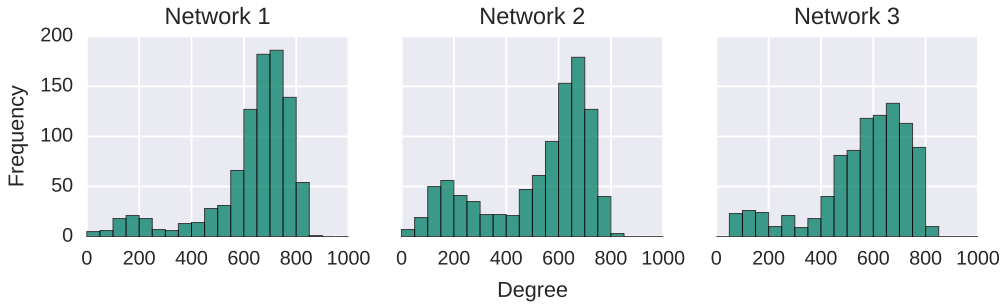


Figure 5.1: Age distribution per network The width of a bar corresponds to two days. For each network bees with a negative age and the queen were removed (11, 10, and 9 bees).

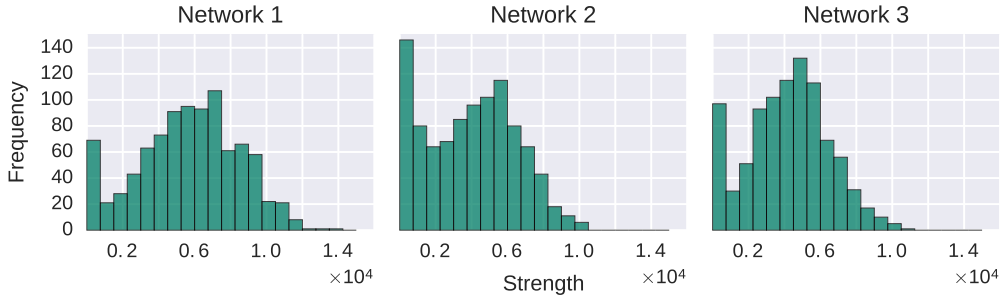
weights is depicted in figure 5.2d.

5.2 Node Level Metrics in Relation to the Age of Bees

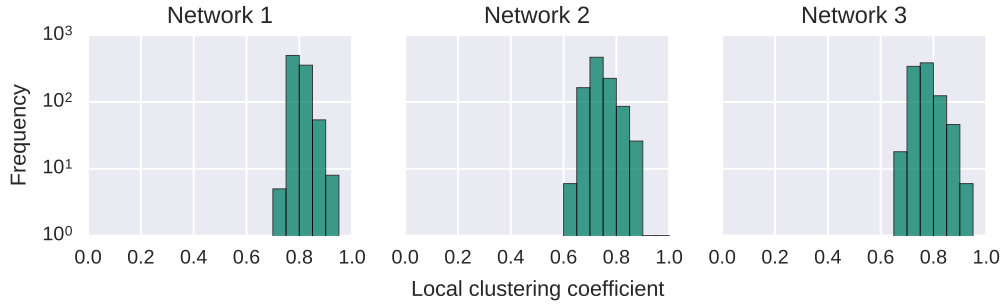
Closenes centrality (weighted and unweighted) in relation to age is depicted in figure 5.7. Betweennes centrality (weighted and unweighted) in relation to age is depicted in figure 5.8. Eigenvector centrality (weighted and unweighted) in relation to age is depicted in figure 5.9.



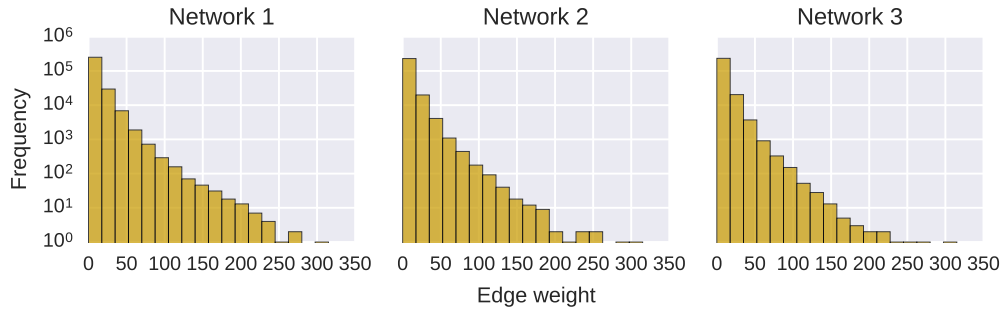
(a) Degree distribution



(b) Strength distribution



(c) Local clustering coefficient



(d) Edge weight distribution

Figure 5.2: Degree, strength and edge weight distribution for all three networks.

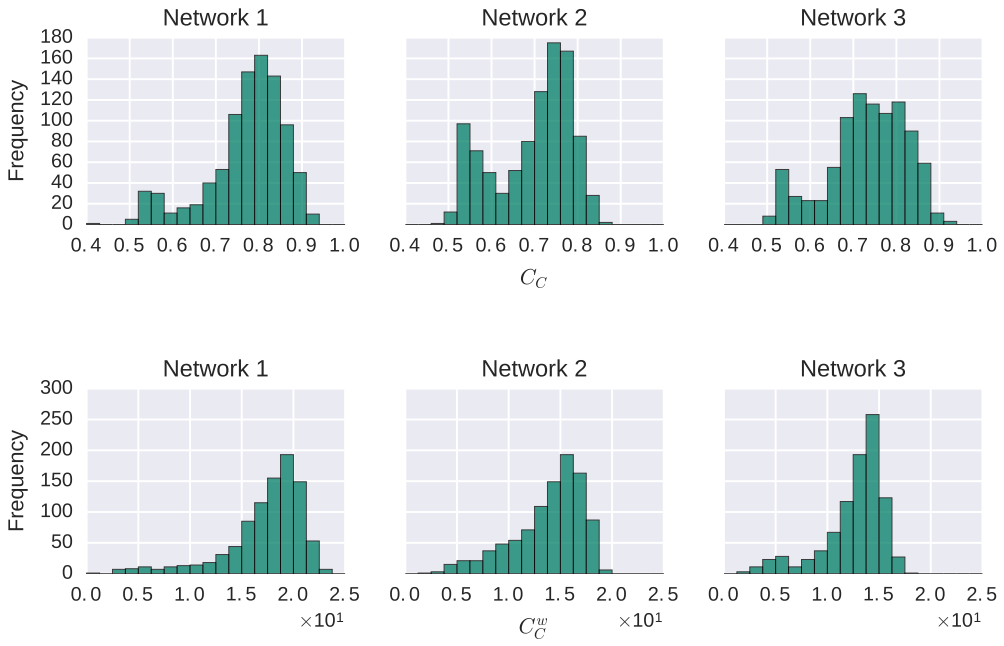


Figure 5.3: Closeness Centrality

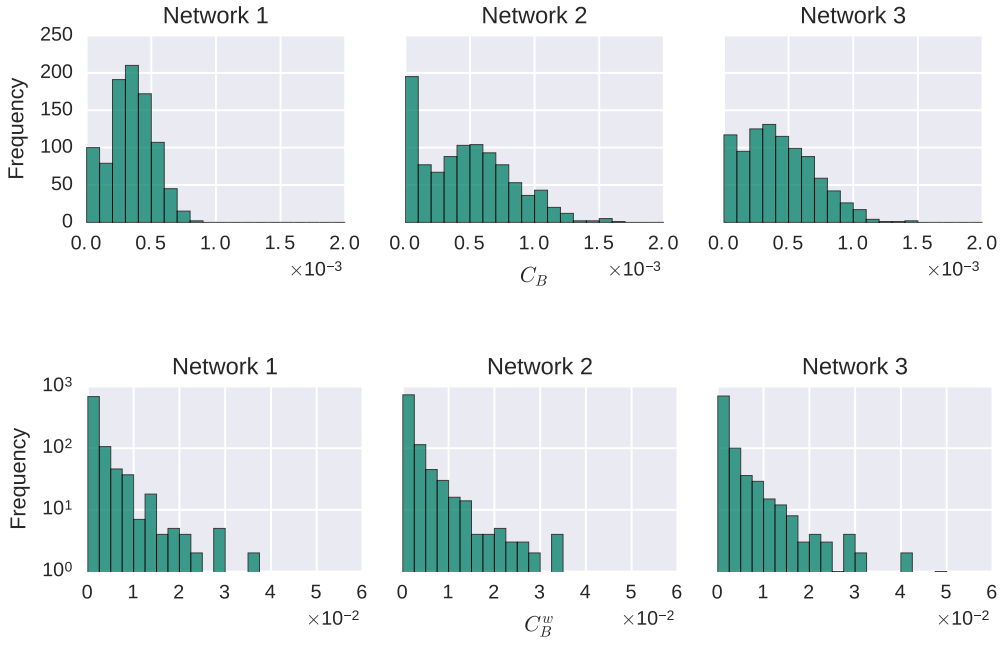


Figure 5.4: Betweenness Centrality

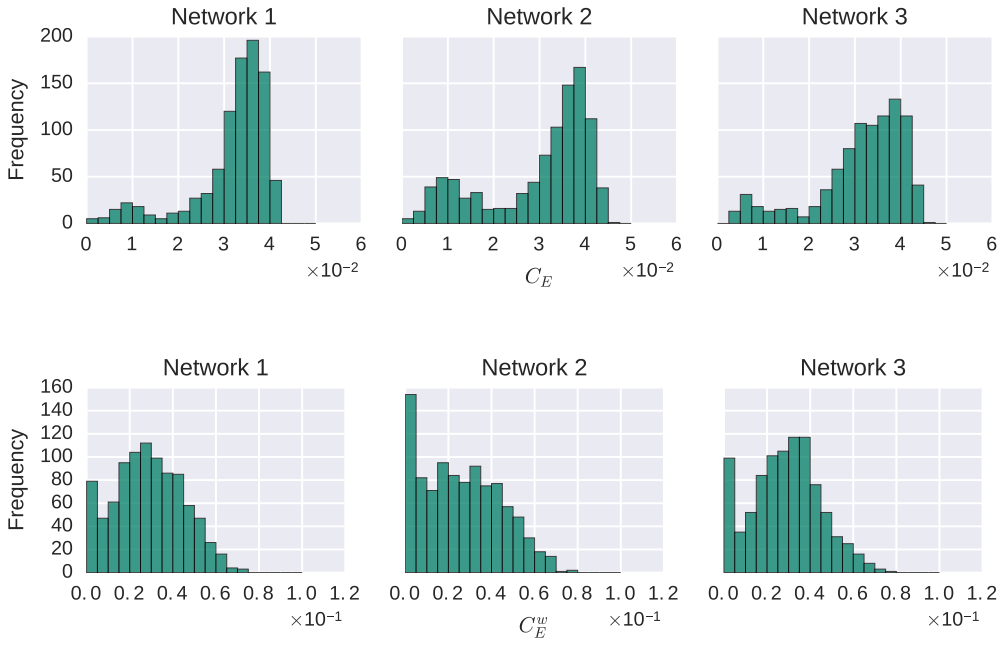


Figure 5.5: Eigenvector Centrality

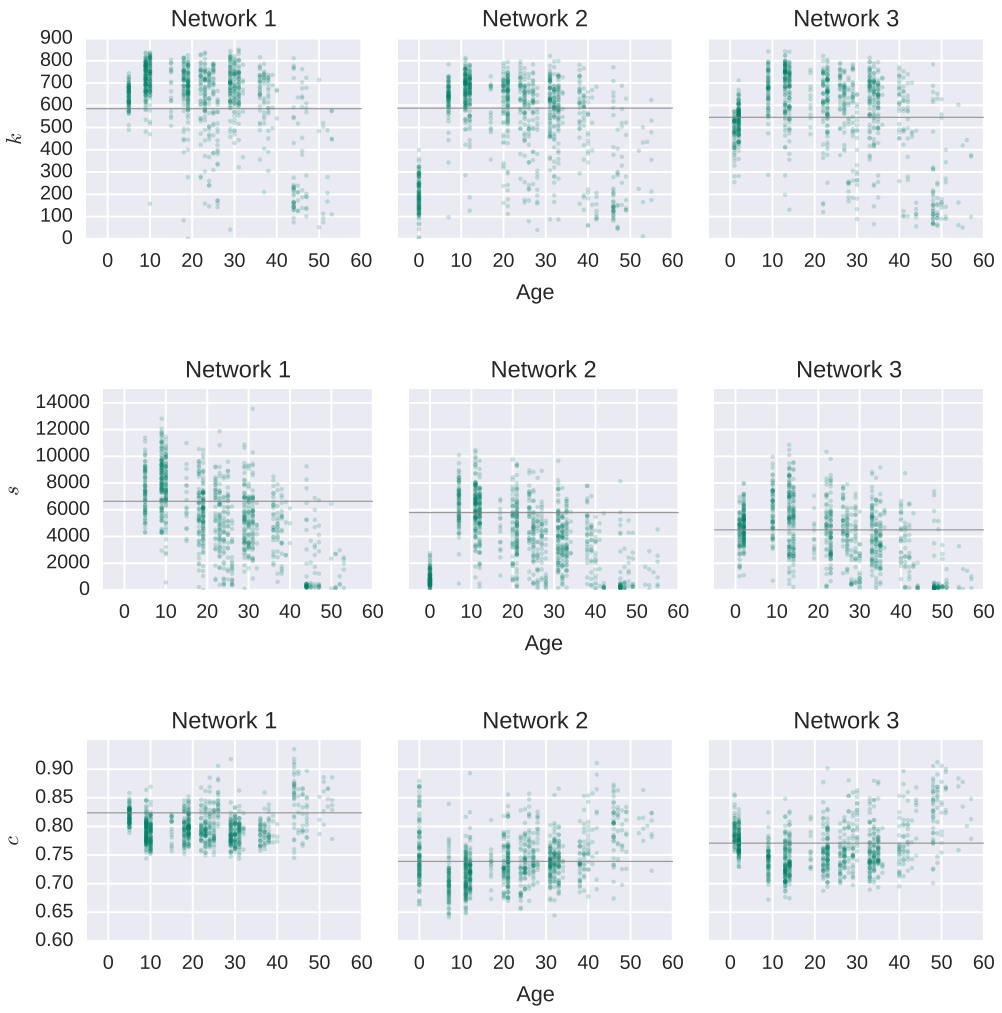


Figure 5.6: Degree, strength and local clustering coefficient in relation to Age The age is measured in days. The gray line corresponds to the quee. Bees with negative age are excluded.

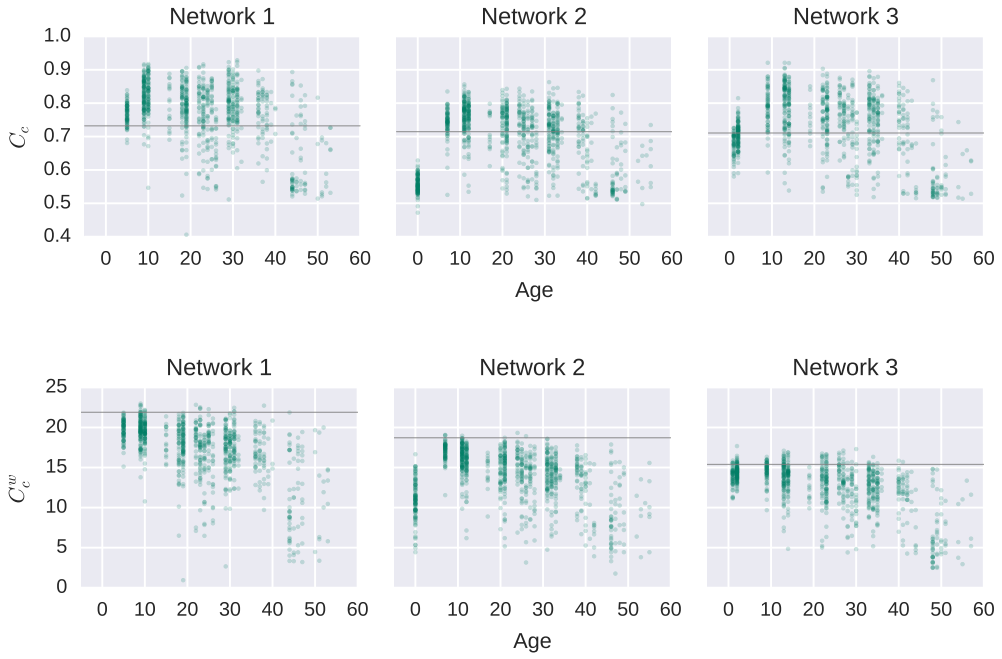


Figure 5.7: Closeness Centrality and Age The age is measured in days. The gray line corresponds to the quee. Bees with negative age are excluded.

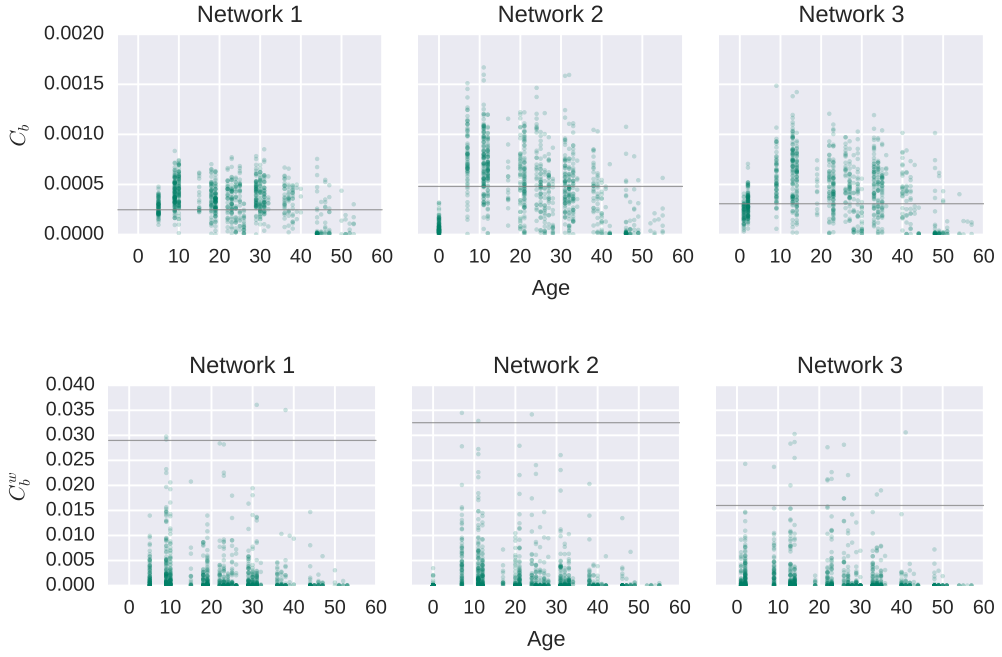


Figure 5.8: Betweennes Centrality and Age The age is measured in days. The gray line corresponds to the quee. Bees with negative age are excluded.

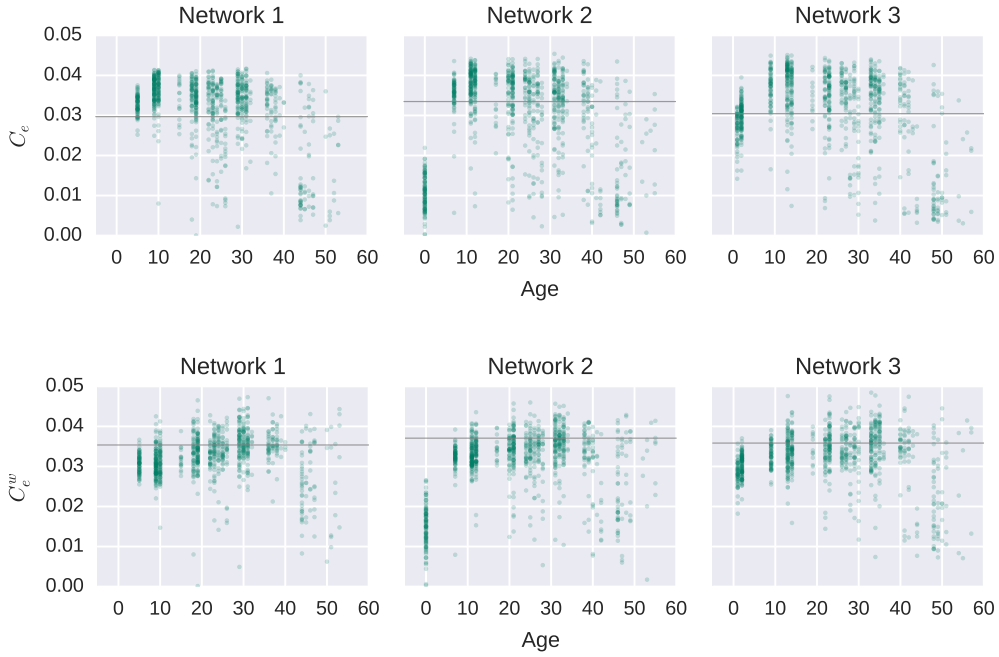


Figure 5.9: Eigenvector Centrality and Age The age is measured in days. The gray line corresponds to the quee. Bees with negative age are excluded.

5.3 Community Detection

Using the leading eigenvector (LE) algorithm in all three networks two communities with about the same number of nodes are detected. With walktrap in network 1 two communities and in network 2 and 3, three communities. The exact number of community members for algorithm is shown in table 5.1.

5.3.1 Age Distribution of Communities

The first community (LE-C1, WT-C1) contains the queen and bees who are on average younger than the second community (LE-C2, WT-C3). For walktraps third community it is a middle-aged community (WT-C2). The age difference for network 1 is 8.4 days, for network 2 10.9 days, and for network 3 14.4 days on average for leading eigenvector communities.

The age distribution for each community and network is depicted in figure 5.10.

A two sample Kolmogorov–Smirnov test showed, that for leading eigenvector communities, the age distributions are significantly different ($p < 0.001$). For walktrap C1 with C2 and C3 are significantly different, but C2 and C3 are not that much significant. Exact p-values are shown in table 5.4

5.3.2 Spatial Distribution of Communities

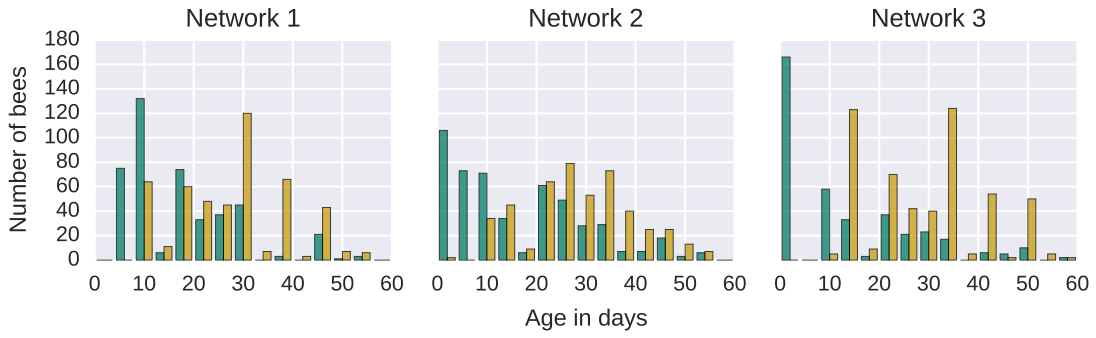
The two communities detected by leading eigenvector are located in two different regions of the honeycomb. The older community (*orange* in figure 5.11)) is in all three networks closer to the hive exit and the younger community (*green* in figure 5.11)) is situated in the comb center. Walktrap revealed the same two communities like leading eigenvector for all three networks, with the same spatial distribution. For network 2 and 3, a third community (*gray* in figure 5.12)) is located between the young and old community.

Table 5.3: Overview about communities per network Communities marked with * contain the queen. Age and standard deviation (SD) are measured in days.

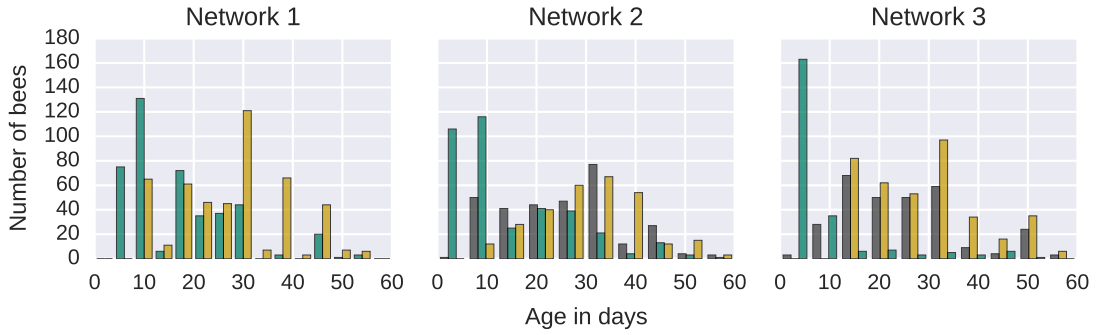
	ID	Members	Proportion	Age	SD
Leading eigenvector					
Network 1	C1	∗434	47.07%	16.81	±17.91
	C2	488	52.93%	25.15	±19.49
Network 2	C1	∗503	51.43%	15.44	±19.54
	C2	475	48.57%	26.37	±18.01
Network 3	C1	∗385	41.76%	12.85	±20.24
	C2	537	58.24%	27.26	±17.84
Walktrap					
Network 1	C1	∗431	46.75%	16.76	±17.92
	C2	490	53.15%	25.16	±19.48
Network 2	C1	∗372	38.04%	12.98	±19.00
	C2	311	31.80%	23.11	±19.48
	C3	294	30.06%	28.15	±16.77
Network 3	C1	∗231	25.05%	7.09	±19.60
	C2	301	32.65%	23.83	±17.22
	C3	390	42.30%	27.63	±18.48

Table 5.4: Kolmogorov-Smirnov test p-values for leading eigenvector (LE) and walktrap (WT) for each network and its communities.

		LE p-value	WT p-value
Network 1	C1, C2	5.1e-32	3.3e-31
Network 2	C1, C2	7.6e-38	2.3e-32
	C1, C3		1.3e-44
	C2, C3		6.6e-05
Network 3	C1, C2	1.4e-64	1.8e-65
	C1, C3		3.9e-93
	C2, C3		2.6e-05

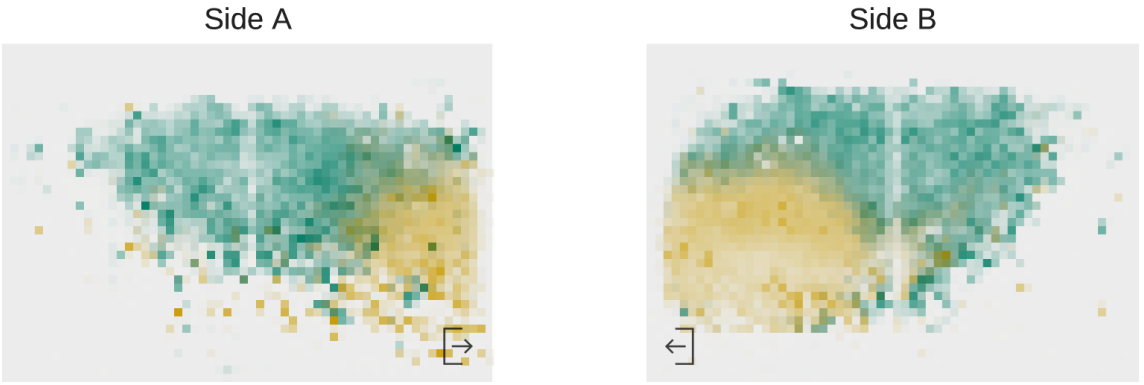


(a) Leading eigenvector

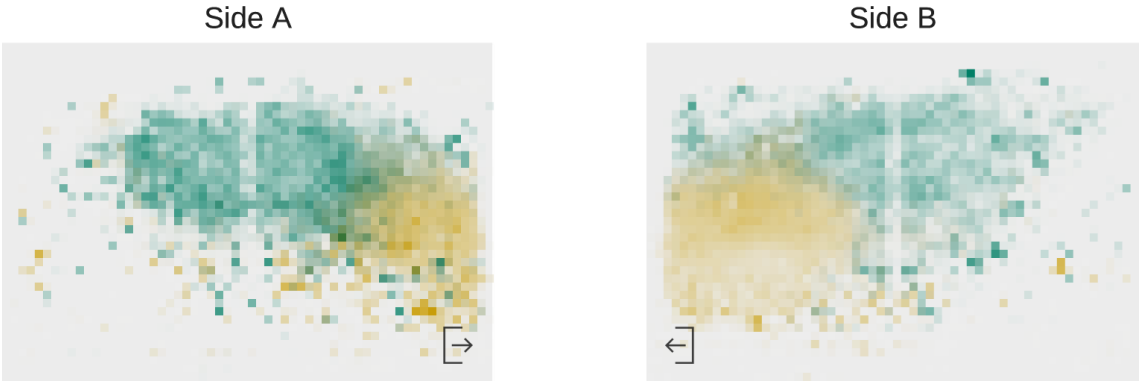


(b) Walktrap

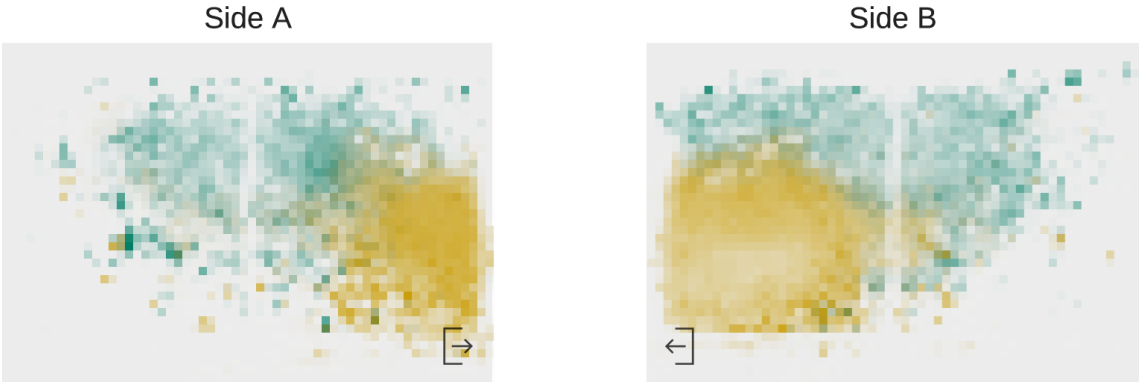
Figure 5.10: Age distribution for each community and network The *green* bar is the community containing the queen. The queens age is not included in the statistic. The *orange* bars coresspond to the second community, containing older bees. The *gray* bars is a third community only revealed by walktrap and contains middle-aged bees.



(a) Network 1

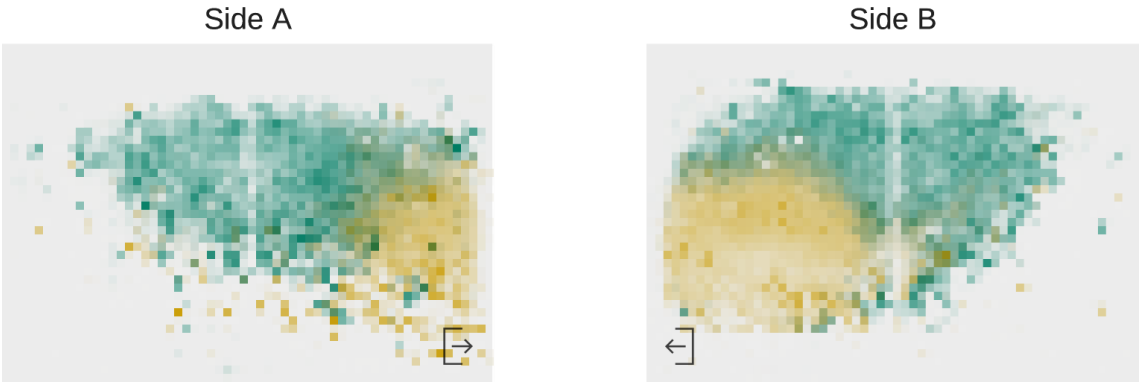


(b) Network 2

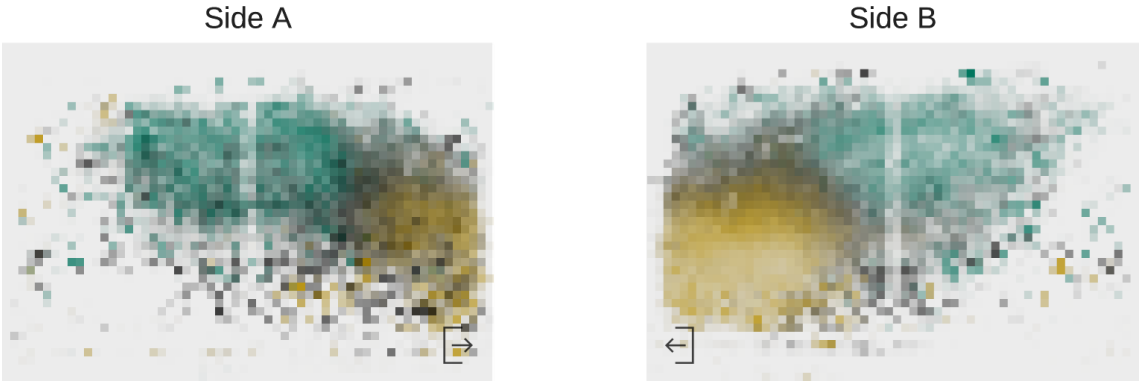


(c) Network 3

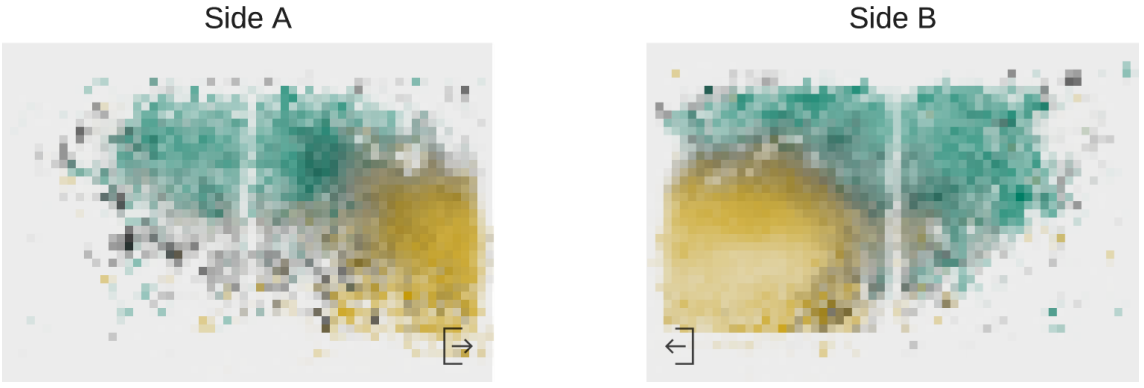
Figure 5.11: Communities per network - leading eigenvector The *green* colour represents the younger community, containing the queen. The *orange* color represents the older community. The hive exit on side A is on the bottom right and on side B on the bottom left. The data is aggredated for the complete timeframe of ten hours.



(a) Network 1



(b) Network 2

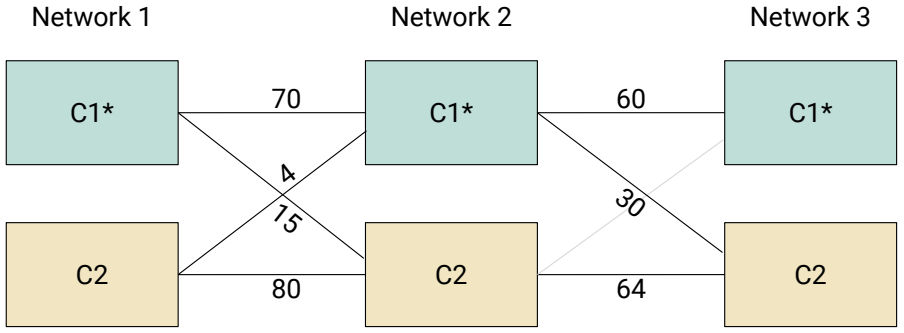


(c) Network 3

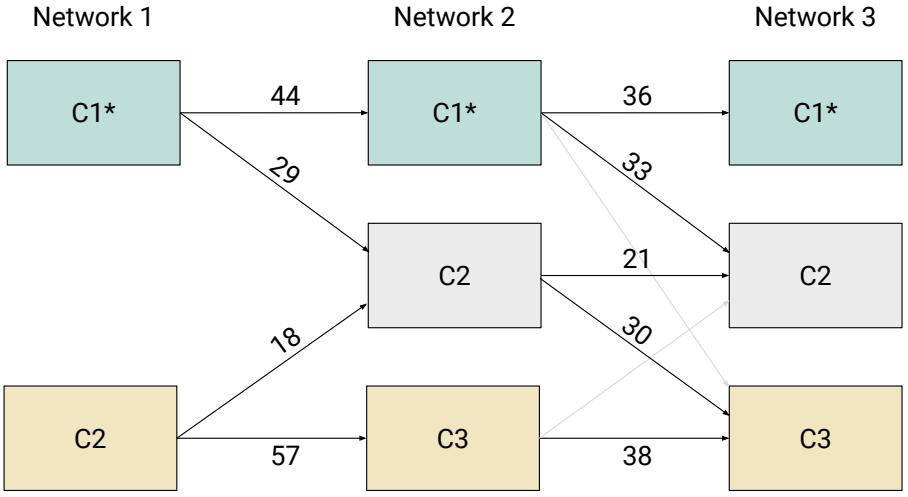
Figure 5.12: Communities per network - walktrap The *green* colour represents the younger community, containing the queen. The *orange* color represents the older community. The *gray* represents the middle-age community. The hive exit on side A is on the bottom right and on side B on the bottom left. The data is aggredated for the complete timeframe of ten hours.

5.4 Community members over time

The match value between the two communities in successive time steps are calculated with formula (2.1) and presented in figure 5.13a for the resulting communities using the leading eigenvector algorithm and figure 5.13b for the communities detected with walktrap.



(a) Leading eigenvector communities



(b) Walktrap communities

Figure 5.13: Community matching The numbers indicate the match values in percent. The community marked with * contains the queen. *Green* represents the younger community *orange* the older community, and gray the middle-aged community. The light gray arrow represents match values below three percent.

Chapter 6

Conclusion

6.1 Discussion

http://mathoverflow.net/questions/126704/the-probability-distribution-for-vertex-

6.2 Limitation

contact duration and contact frequency
distances is are hard to measure, because hive is very small and dense (is spatial proximity a good proxy in this case)

nur weil man mehr daten hat, muss es nicht unbedingt besser werden
aber mehr daten (aufbereiteter guter datensatz), evtl. fragen stellen, an die man vorher noch nicht gedacht hat.

6.3 Summary and Future Work

Measuring the robustness of network community structure using assortativity [28]

Bibliography

[1] Thomas Aynaud et al. “Communities in evolving networks: definitions, detection, and analysis techniques”. In: *Dynamics On and Of Complex Networks, Volume 2*. Springer, 2013, pp. 159–200.

[2] Albert-László Barabási. *Network science*. Cambridge University Press, 2016.

[3] David Baracchi and Alessandro Cini. “A Socio-Spatial Combined Approach Confirms a Highly Compartmentalised Structure in Honeybees”. In: *Ethology* 120.12 (2014), pp. 1167–1176.

[4] Alain Barrat et al. “The architecture of complex weighted networks”. In: *Proceedings of the National Academy of Sciences of the United States of America* 101.11 (2004), pp. 3747–3752.

[5] Benjamin Blonder and Anna Dornhaus. “Time-ordered networks reveal limitations to information flow in ant colonies”. In: *PloS one* 6.5 (2011), e20298.

[6] Benjamin Blonder et al. “Temporal dynamics and network analysis”. In: *Methods in Ecology and Evolution* 3.6 (2012), pp. 958–972.

[7] Piotr Bródka, Stanisław Saganowski, and Przemysław Kazienko. “Community Evolution”. In: *Encyclopedia of Social Network Analysis and Mining* (2014), pp. 220–232.

[8] Daniel Charbonneau, Benjamin Blonder, and Anna Dornhaus. “Social insects: a model system for network dynamics”. In: *Temporal Networks*. Springer, 2013, pp. 217–244.

[9] James D Crall et al. “BEEtag: a low-cost, image-based tracking system for the study of animal behavior and locomotion”. In: *PloS one* 10.9 (2015), e0136487.

[10] Darren P Croft, Richard James, and Jens Krause. *Exploring animal social networks*. Princeton University Press, 2008.

[11] Damien R Farine and Hal Whitehead. “Constructing, conducting and interpreting animal social network analysis”. In: *Journal of Animal Ecology* 84.5 (2015), pp. 1144–1163.

[12] Mark Fiala. “Comparing artag and artoolkit plus fiducial marker systems”. In: *Haptic Audio Visual Environments and their Applications, 2005. IEEE International Workshop on*. IEEE. 2005, 6–pp.

[13] Santo Fortunato. “Community detection in graphs”. In: *Physics reports* 486.3 (2010), pp. 75–174.

[14] Steve Harenberg et al. “Community detection in large-scale networks: a survey and empirical evaluation”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 6.6 (2014), pp. 426–439.

[15] John Hopcroft et al. “Tracking evolving communities in large linked networks”. In: *Proceedings of the National Academy of Sciences* 101.suppl 1 (2004), pp. 5249–5253.

[16] Raphaël Jeanson. “Long-term dynamics in proximity networks in ants”. In: *Animal Behaviour* 83.4 (2012), pp. 915–923.

[17] Jens Krause et al. *Animal social networks*. Oxford University Press, USA, 2014.

[18] Danielle P Mersch, Alessandro Crespi, and Laurent Keller. “Tracking individuals shows spatial fidelity is a key regulator of ant social organization”. In: *Science* 340.6136 (2013), pp. 1090–1093.

[19] James Moody, Daniel McFarland, and Skye Bender-deMoll. “Dynamic network visualization 1”. In: *American journal of sociology* 110.4 (2005), pp. 1206–1241.

[20] Dhruba Naug. “Structure of the social network and its influence on transmission dynamics in a honeybee colony”. In: *Behavioral Ecology and Sociobiology* 62.11 (2008), pp. 1719–1725.

[21] M. Newman. *Networks: An Introduction*. OUP Oxford, 2010.

[22] Mark EJ Newman. “Finding community structure in networks using the eigenvectors of matrices”. In: *Physical review E* 74.3 (2006), p. 036104.

[23] Michael C Otterstatter and James D Thomson. “Contact networks and transmission of an intestinal pathogen in bumble bee (Bombus impatiens) colonies”. In: *Oecologia* 154.2 (2007), pp. 411–421.

[24] Gergely Palla et al. “Uncovering the overlapping community structure of complex networks in nature and society”. In: *Nature* 435.7043 (2005), pp. 814–818.

[25] Noa Pinter-Wollman et al. “The dynamics of animal social networks: analytical, conceptual, and theoretical advances”. In: *Behavioral Ecology* 25.2 (2014), p. 242. eprint: `/oup/backfile/Content_public/Journal/beheco/25/2/10.1093/beheco/art047/2/art047.pdf`.

[26] Pascal Pons and Matthieu Latapy. “Computing communities in large networks using random walks”. In: *International Symposium on Computer and Information Sciences*. Springer. 2005, pp. 284–293.

[27] Lauren E Quevillon et al. “Social, spatial, and temporal organization in a complex insect society”. In: *Scientific reports* 5 (2015).

[28] Daizaburo Shizuka and Damien R Farine. “Measuring the robustness of network community structure using assortativity”. In: *Animal behaviour* 112 (2016), pp. 237–246.

[29] Leon Sixt. “RenderGAN: Generating realistic labeled data - with an application on decoding bee tags”. B.S. Thesis. Freie Universtät Berlin.

[30] Fernando Wario et al. “Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees”. In: (2015).

[31] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*. Vol. 8. Cambridge university press, 1994.

[32] Tina Wey et al. "Social network analysis of animal behaviour: a promising tool for the study of sociality". In: *Animal behaviour* 75.2 (2008), pp. 333–344.

[33] Zhao Yang, René Algesheimer, and Claudio J Tessone. "A Comparative Analysis of Community Detection Algorithms on Artificial Networks". In: *Scientific Reports* 6 (2016).

List of Figures

List of Tables

Appendix A

Appendix Stuff

[TODO: Table with studies related to social insects and bees and manual and automatic tracking.]

[TODO: Figure: Maintance and Faliour Days, also maybe timeline for each camera]

[TODO:]