UNIVERSITY OF CALIFORNIA,
IRVINE

Route Clustering for Strategic Planning in Air Traffic Management

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Mechanical and Aerospace Engineering

by

Adrià Segarra Torné

Thesis Committee:
Professor Kenneth D. Mease, Chair
Assistant Professor Solmaz S. Kia
Assistant Professor Haithem E. Taha

2015

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I would like to start by expressing my honest gratitude to Pete J. Balsells, the University of California, Irvine and the ‘Generalitat de Catalunya’. This work would not have been possible without the support of the Balsells Graduate Fellowship Program.

I also want to show my gratitude to the University Affiliated Research Center at NASA Ames. The UARC has been an essential institution for this project, allowing me to become an intern at NASA. This thesis would not have been possible without the opportunity and support they provided.

I would like to thank the advisor of this thesis and chair of the Mechanical and Aerospace Engineering Department at UC Irvine, Prof. Kenneth D. Mease, for being a critical and constructive voice during the development of this project. His advice and constant attention to the global problem being faced has provided the necessary coherence to this work.

A very special mention must be addressed to Alessandro Bombelli, PhD candidate at UC Irvine, for developing much of the previous work in this area. He was crucial in helping me understand the details of the Aggregate Model, and a very valuable source of help and advice. He also became a good friend, and made my experience at UC Irvine more enjoyable.

My sincere gratitude goes to Dr. Heather Arneson and Dr. Anthony Evans from NASA Ames Research Center for their key role in this thesis. They have guided me and have been a reliable source of Air Traffic Management and Machine Learning knowledge. I thank them for the daily discussions and frequent revisions of my advances, and for being the most important inspiration and motivation to continue improving my work. They have become good friends and helped me make the most out of my stay at NASA Ames.

I thank Dr. Eric Trumbauer from UC Irvine for his advice during several discussions. He has helped ensure the mathematical correction of various points of the project.

I would like to thank Prof. Roger H. Rangel from UC Irvine, director of the California-Catalonia Programs, for his help during the last two years.

Last but not least, I would like to thank my family for all their support and motivation throughout this journey. Despite being far, they have been a vital part of it.

This material is based upon work supported by the National Aeronautics and Space Administration under Prime Contract Number NAS2-03144 awarded to the University of California, Santa Cruz, University Affiliated Research Center.

ABSTRACT OF THE THESIS

Route Clustering for Strategic Planning in Air Traffic Management

By

Adrià Segarra Torné

Master of Science in Mechanical and Aerospace Engineering

University of California, Irvine, 2015

Professor Kenneth D. Mease, Chair

The volume of air traffic in the National Air Space has been growing at a very fast pace during recent years, and increasing demand for air travel in coming years is predicted. Strategic Planning is a necessary tool to guarantee a safe and efficient increase of Air Traffic. An Aggregate Model for Strategic Planning of Air Traffic Management is the framework of this project.

Aggregate models require the creation of a network that is representative of the expected flow and which will serve as a platform to solve flow optimization problems. We have integrated an automatic method for route clustering – S_{max} method – to generate the required network. Automatic clustering is necessary in order to efficiently cluster many individual scenarios.

Some available alternatives to determine the number of clusters are tested. The studied alternatives provide success rates that oscillates between 49% and 68%. In addition to the relatively low success rates, these methods require one user-input parameter, and they are highly sensitive to it.

A different method, based on the Silhouette Score and the Dip Test measures, is developed. The S_{max} method requires no user-input parameters, and it consistently provides rates of success that approximately oscillate between the 72% and the 81%.

The presented clustering approach noticeably improves the rate of correct clustering cases for the specific scenario of route clustering.

Chapter 1

Aim of the Project

The aim of this project is to develop an automatic route clustering technique in order to improve the quality of the resulting aggregate network which can be used as a platform to solve air traffic flow optimization problems.

Chapter 2

Scope of the Project

- Determine the adequate characteristics of the dataset to generate the network.
- Classify flights from the dataset in homogeneous groups.
- Define the dissimilarity metric.
- Identify outlier flights.
- Study the available clustering methods. Modify as necessary.
- Analyze the performance of the final clustering method.

Chapter 3

Background in Air Traffic Management

3.1 Classical Tools and Organization of ATM

Based on [1] we can determine the basic features of the current Air Traffic Flow Management (ATFM) model. This will help the reader understand the context where Strategic Planning would be useful, and how it would be implemented.

3.1.1 Organization and Structure

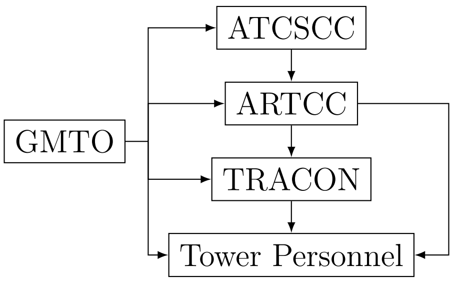

Figure 3.1: Hierarchy in ATM

- **GMTOs** (General Managers of Tactical Operations): provide oversight and line authority to Traffic Management Personnel. Expert Air Traffic Control (ATC) advisors. Provide daily updates on ATM initiatives.

- **ATCSCC** (David J. Hurley Air Traffic Control System Command Center): Maximum authority in ATC/ATM in the National Airspace System (NAS). Responsible for Air Traffic Flow Management.

- **ARTCCs** (Air Route Traffic Control Centers): Control the aircraft in its specific airspace, which is further divided into sectors.

- **TRACON** (Terminal Radar Approach Control): Control the aircraft in the terminal airspace (5 to 40 miles from airport, or up to 10,000 feet).

- **Tower Personnel**: Give departure clearance, control aircraft on the ground and within 5 miles.

A distinction must be made between Air Traffic Control (ATC) and Air Traffic Management (ATM). The management decisions (strategic planning) are taken centrally at the ATCSCC, whereas the control decisions (to assure separation between aircraft and safety in the operations) are taken locally at ARTCCs and TRACONs (and ultimately through tower personnel). The outputs of the management function are traffic management initiatives (TMIs), also called traffic flow management (TFM) initiatives, which are implemented through the *control* side.

3.1.2 ATC/ATM tools

It is necessary to present the current tools that are being used to manage the NAS. An understanding of these tools will provide a framework for the future Strategic Planning model.

- **Sequencing Programs**: designed to achieve a specified interval between aircraft (to assure a safe minimum separation).

- **Altitude Segregation**: used to separate flows of traffic or to distribute the number of aircraft that require access to a specific area. *Low Altitude Alternate Departure Routing, Capping* and *Tunneling* are the main examples of these tools.

- **Ground Delay Programs (GDP):** procedure where aircraft are delayed at their departure airport in order to manage demand and capacity at their arrival airport, or in support of Sever Weather Avoidance Plan (SWAP).

- **Ground Stops (GS)**: procedure requiring aircraft that meet a specific criteria to remain on the ground. They may be caused by severe weather, equipment outages, catastrophic events, saturated sectors, and others.

- **Airspace Flow Programs (AFP)**: provide enhanced en-route traffic management during severe weather events. It will automatically assign new EDCTs (departure times) to those aircraft whose route would be affected by severe weather, in order to avoid this weather (causing delays). Airborne holding and rerouting can be applied if approved by ARTCC or TRACON, but are not considered by the program. In case rerouting is applied, the new route is usually proposed by the user/airline.

- **Flight Schedule Monitor (FSM)**[2]: simulation/modeling tool used in the NAS in support of GDPs, GS and AFPs (also airborne holding, if it must be planned). The inputs for this tool are the scheduled flights information during a specific time-frame, as well as some hypothetical TMIs if demand is found to be greater than capacity (TMIs are GDPs, GS, AFPs and airborne holding). The outputs of this tool will be specific flight information, arrival and departure rates, open arrival slots and other pertinent traffic flow information. FSM provides a graphical and time-line presentation of airport and airspace demand and capacity, and this information is used by ATCSCC to plan the necessary TMIs. The FSM has the capability to implement TMIs to balance demand and capacity on airports and airspace.

- **Time Based Flow Management (TBFM)**: additional tool to adjust capacity/demand imbalances at select airports and en-route points across the NAS.

- **Traffic Management Advisor (TMA)**: comprehensive automated tool for planning efficient flight trajectories from cruise altitude to the runway threshold.

- **Adaptive Compression (AC)**: helps ensure that all slots in a program are used.

- **Integrated Collaborative Rerouting (ICR)**: used to reroute aircraft around en-route constrains, incorporating operator preferences where possible.

- **North American Route Programs (NRP)**: specifies provisions for flight planning at flight level 290 (FL290) and above, within the conterminous U.S. and Canada.

- **Special Traffic Management Program (STMP)**: long-range strategic initiative that is implemented when a location requires special handling to accommodate above-normal traffic demand, e.g. a sports event.

The communication of all TMIs and important information is mainly made in two ways: the Operational Information System (OIS) and the Advisories. The OIS is a website that contains various relevant information, while the Advisories are distributed electronically when necessary.

The Planning Team in charge of these tools is composed of FAA personnel at the ATCSCC. The ATCSCC hosts a planning telephone conference every two hours to identify any constraints to the NAS for the next six hours. The team members present their ideas and concerns and develop an Operations Plan that explains the constraints and how they will be managed. A tool to aid Strategic Planning would be highly beneficial for this team.

3.1.3 Weather Tools

The main weather tools that are currently used by Traffic Management personnel are the following:

- **Terminal Area Forecasts (TAF)**: describe anticipated weather conditions at airports. In the U.S, these forecasts are produced every eight hours by the National Weather Service (NWS).
- Convective outlooks forecast the most severe thunderstorms in the U.S. for the next 18 hours. They are updated several times throughout the day.
- The **Collaborative Convective Forecast Product (CCFP)** is a forecast for intense convection activity made for two-, four- and six-hour periods by a group consisting of the NWS, the aircraft operators, ARTCC weather units and the meteorological service of Canada.

Severe Weather Avoidance Plan (SWAP)

SWAP is a formalized program that is developed for areas susceptible to disruption in air traffic flows caused by thunderstorms. Each air traffic facility may develop its own strategy for managing the severe weather event. Their plan then becomes part of the overall daily operations plan. As each weather event is unique, the response is tailored to meet the specific forecasted and actual events of the day. The SWAP plan is issued through the Planning Team.

3.1.4 Routes

The routes to be used are mainly taken from one of the following sources:

- **National Playbooks**: collection of SWAP routes that have been pre-validated.
- **Preferred Routes**: routes that are requested, and have been published by ATC to inform users of the normal traffic flows between airports.
- **Coded Departure Routes (CDR)**: combination of coded air traffic routings and refined coordination procedures.

Each source is used in different scenarios, and the routes are selected by ATC personnel or requested by the user.

3.2 NextGen

The *Next Generation Air Transportation System*, or NextGen, is an upgrade on the U.S. air transportation system that aims at maintaining or increasing the safety in the operations, while allowing the predicted growth in air traffic in the upcoming years. More information can be found in the references of [3].

3.3 Justification

The current ATFM model is the result of successive additions to the first existing tools when commercial aviation appeared. Looking at the presented tools we can identify a clear decentralization on the tasks performed: there is a tool for each task. In addition, these tools are mainly focused on solving problems during brief periods of time in localized regions of airspace, what is known as *tactical planning*.

Tactical planning is very necessary due to the uncertainty of the factors determining the evolution of the air traffic. Nevertheless, a better overall organization of the air traffic flow over bigger regions of airspace could help reduce the use of tactical planning tools. The kind of planning that affects great domains of airspace and deals with longer planning horizons (2-8h) is known as *strategic planning.* Strategic planning as an automatized tool to aid human decision-makers is increasingly necessary as air traffic grows in the National Air Space.

The context of this work is the development of a centralized tool to asses the strategic planning in the whole National Air Space, accounting for the appropriate restrictions and limitations and including the weather effects. Our focus is the creation of the aggregate network that would be used in this optimization problem.

A global strategic planning must be integrated with the currently used tools and resources in order to be a useful aid. In the context of the network creation, this involves a correct choice of the dataset, which will be discussed in detail later, and an appropriate representation of the flows by the aggregate network, consisting in routes that controllers would identify as main flows and therefore use.

For the purposes of the model evaluation, only historical flight tracks have been used. In a real application, this could be enriched by adding the routes from the alternative sources mentioned in *Subsection 3.1.4 Routes.* This is not a crucial step, though, as the historical flight tracks should already include most of the routes represented in these alternative sources, for a big enough dataset. The addition or omission of this additional routes may have operational effects, but it has no effect on the clustering method study that will be discussed on this project.

Chapter 4

Previous Work

The work that will be carried out during this thesis is an addition to the existing work of Alessandro Bombelli, Lluís Soler and Prof. Kenneth Mease. Reading of [4] is highly encouraged, as it is a necessary reference to understand the context of the work that will be conducted here. For convenience, the summary of that publication is added:

“The Aggregate Route Model for strategic Traffic Flow Management is presented. It is an Eulerian flow model whose cells are discrete elements of unidirectional point-to-point routes, each cell with the same transit time. The aggregate routes are determined from flight data based on similarity measures. Spatial similarity is judged by the Fréchet distance and temporal similarity by average speed. The traffic controls accounted for in the model are ground delays and pre-departure reroutes. The resulting traffic flow network is then translated into a discrete linear time-invariant system. Centralized strategic traffic flow planning is posed as a linear programming problem. The total delay is minimized subject to sector capacity constraints. Two examples demonstrate the planning: in the first, ground delays are adjusted to plan traffic in the Los Angeles Center, and in the second, ground delays and pre-departure reroutes are planned to manage a scenario with convective weather impeding departures from the Dallas Fort Worth airport."

The work presented here constitutes a change on how the Aggregate Routes for this model are created. We aim at improving both the automation of the process and the quality of the resulting network.

Chapter 5

Clustering Overview

5.1 Clustering Framework

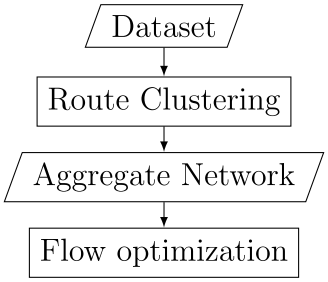

Figure 5.1: Main program

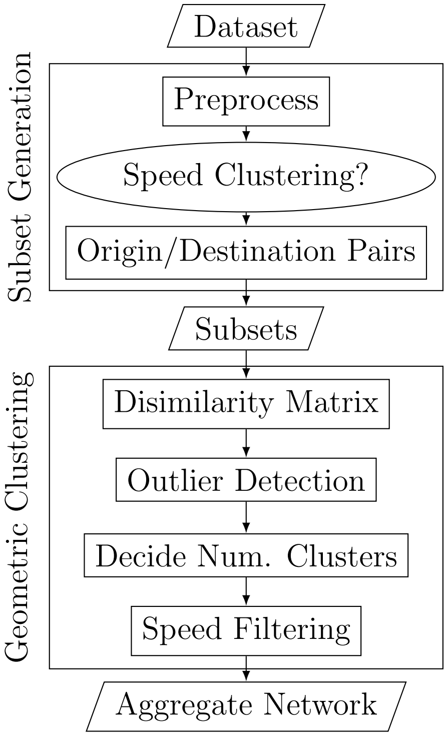

Figure 5.2: Route clustering process

In Figure 5.1 the main organization of the global program is presented to provide context. The dataset (flight tracks) is clustered, obtaining an aggregate network. This aggregate network will serve as an input to solve the flow optimization problem. In this project we will focus on the route clustering process, which we present in Figure 5.2.

The first steps of the clustering process can be understood as the subset generation. The clustering process starts with the dataset consisting on files containing the information of interest of all flights in the national airspace. Each file contains information corresponding to all flights during one specific day. Once the days of interest have been chosen, the raw text files (known as TRX files) are preprocessed and transformed into a format better suitable to be read afterwards. This step noticeably increases the speed and performance of the process.

Then the flights of interest are evaluated to determine the need for speed clustering. If clustering is found suitable, two groups are formed based on their speed.

The next step is to further divide the dataset into subgroups of flights that share the same origin and destination airport. The resulting groups of flights are called subsets, and the combination of all subsets is equivalent to the original dataset. The following steps can be grouped into what we will call geometric clustering. First the similarity matrix for each subset must be calculated. It contains the pairwise distances for each pair of routes in a subset. Once the pairwise distances are known, outliers must be detected, and the number of clusters for each subset decided. Then the flights with speeds that are too dissimilar compared to the other flights in their cluster will be discarded, to provide further dynamic consistency to the clusters. Finally the representative route for each cluster (aggregate route) will be obtained as a combination of all routes in that cluster.

Each subset will yield an aggregate subnetwork (combination of aggregate routes in that subset), and the combination of all aggregate subnetworks results in the global aggregate network, which is representative of the initial dataset.

All these steps will be explained in detail, with special emphasis in the geometric clustering.

5.2 What is Geometric Clustering

In this context we will refer to geometric clustering as the formation of groups of routes that are geometrically similar. The correct identification of such groups will allow us to describe the flow between a specific origin/destination pair (O/D pair) using a lower dimension network.

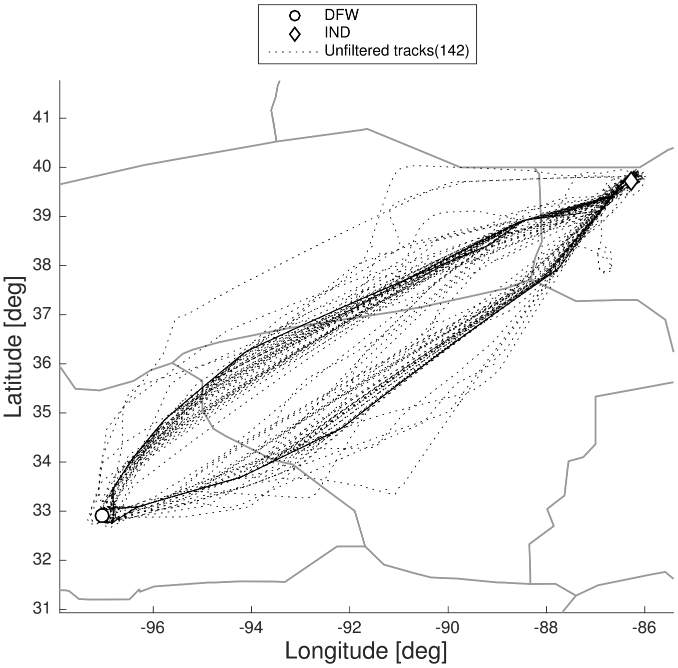

Figure 5.3: Tracks from DFW to IND.

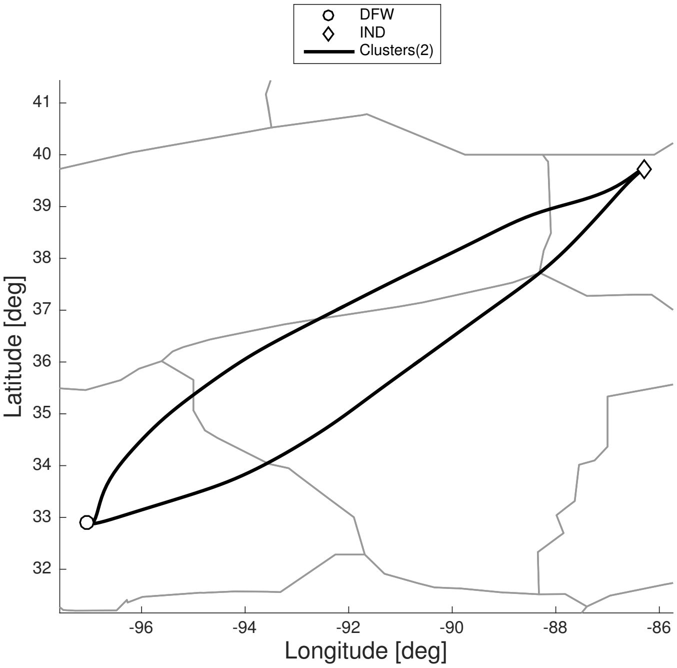

Figure 5.4: Clusters from DFW to IND.

For example, the flow between DFW and PHX airports for a specific timeframe consists of

142 flight tracks, as seen in Figure 5.3. But we can visually identify two distinct regions where most routes are concentrated. Therefore, one could describe the same flow using two appropriate clusters (or aggregate routes) as represented in Figure 5.4.

The advantage of an aggregate representation is the considerable reduction of the network dimension, which is very important when solving flow optimization problems. In the previous example, the original flow of 142 routes is now represented using only 2 clusters. Nevertheless, this approach also has some shortcomings, the most relevant being the inevitable loss of information. It will be important to guarantee that no relevant information is lost.

In the context of a strategic planning approach, the loss of detailed information that clustering implies (if carried out correctly) is considered acceptable, compared to its benefits.

5.3 Need for Automatic Clustering

Clustering processes usually require input parameters that must be tuned manually for each clustering scenario. In our study case, an individual clustering process is required for each origin/destination pair, as we will only cluster flights belonging to the same subset. For n airports, the number of origin/destination pairs (with $A \to B \neq B \to A$, A and B being airports) is

$$N = (n - 1)n \tag{5.1}$$

so clustering must be carried out N times.

Considering only ASPM77 airports in the national airspace, $N = 5,852$. Manual tuning of parameters for all these cases would be a tedious, inefficient and imprecise task, so the need for an automatic clustering method is justified. There are some automatic clustering techniques in the literature which will be discussed in the corresponding section.

Chapter 6

Generating the Subsets

6.1 Dataset

The dataset consists of the multiple flight tracks that will be clustered to generate the aggregate network. The flight tracks in raw format are obtained from *TRX files*, which are *txt* files, each of them containing the relevant information for all flights in the national airspace for one whole day. In addition to the flight tracks, these files contain additional information like the aircraft type, O/D airports or filed flight plan among others. The process by which this information is gathered and organized will be omitted, as it is not relevant for the object of this project.

The first step in the generation of our dataset is to decide the best combination of days to obtain representative clusters. Because the objective of this network is to be used as a platform to solve flow optimization problems in potential contingency scenarios (because of weather, sector capacity, etc.) we must make sure that enough alternatives are represented in our network. The approach we used is to choose the 60 days with worst weather conditions of SWAP season 2014 (April 15th through September 20th). Days with convective weather will contain the characteristic contingency routes being used in those situations. Other contingency scenarios will also be represented in a dataset of this size.

The Weather Impacted Traffic Index (WITI) [5] will be used to quantify the severity of the weather impact on a specific day. WITI is an indicator of the number of aircraft affected by weather. The first step is the generation of a grid that covers the whole national airspace. Then the computation of WITI at an instant of time k (typically at 1-min intervals) is as follows [6],

$$WITI(k) = \sum_{j=1}^{m} \sum_{i=1}^{n} T_{i,j}(k) W_{i,j}(k) \tag{6.1}$$

where $W_{i,j}(k) = 1$ if severe weather is present in the grid element (i, j) at instant k or 0 otherwise, $T_{i,j}(k)$ is the number of aircraft in the grid element (i, j) at instant k, and n and m are the number of rows and columns in the weather grid, respectively. The measure used to choose the worst-weather days is the daily WITI over our domain, which is the addition of all the instant WITI scores during that day. We have chosen the 60 days with the highest daily WITI scores.

The domain of choice for the study case is the whole National Air Space. The reason to choose such a domain is the variety of scenarios it provides. If we want to be able to automatically cluster very different subsets we must include them from an early point in the development of the method.

We will focus on the domestic traffic taking off from Dallas Fort Worth airport (DFW) and directed towards all other ASPM77 airports [7]. This dataset contains a wide diversity of subsets (various length scales, densities and distributions) that is considered to be a good representation of all the cases of interest. The developed method is general and applicable to any given dataset and domain.

In addition to choosing the specific days of interest and the domain of study, the time of the day is considered to be a relevant factor as well. During the night air traffic is, in general, very low, and flights are allowed to take direct routes more often. In case weather must be avoided, the deviations will be as small as possible. This is due to the lack of sector capacity constraints and operational limitations. Including night flights (local time) in our dataset will cause a heavy imbalance, resulting in a clear increase in route usage for the most direct routes. This effect is similar to the one caused by including weather-free days in our dataset. As we will see later on, we want to minimize the cluster density variations as much as possible, and therefore only flights taking off and landing between 9am and 9pm (central time) will be considered. Only complete flights are included in our dataset.

The final dataset consists of 17,111 flights departing from DFW to other ASPM77 airports, taking off after 9am and landing before 9pm. The average departure rate is 24 aircraft/h. It must be noticed that this corresponds only to departures to other ASPM77 airports, so the total average departure rate including all departures for DFW will be higher.

Once our dataset is formed, it is important to ensure that we only cluster flights that are similar in all aspects, as they will eventually be represented by one single route. Therefore, the more variability there is in the features of the clustered individual routes, the higher the error when representing them by one single aggregate route. We must accept some variability in order to cluster, but we also want to ensure that the similarity of features within routes is reasonable.

There are three kinds of similarity that we want to guarantee:

- Speed similarity.
- Operational similarity.
- Geometric similarity.

In the following sections we will discuss how speed similarity and operational similarity are guaranteed, resulting in the final subsets of flights. The next step, geometric similarity (through geometric clustering), is the core of this work and will be treated in several chapters.

6.2 Speed Similarity

The first step to guarantee speed similarity is to calculate the average cruise speed for each flight in our dataset. Next, if our dataset contains flights originating from more than one airport, it is divided into smaller datasets, each of them corresponding to all flights originating form the same airport. This is done because the following speed clustering will be more flexible if applied to single airports than to the whole dataset.

Then, using the *k-means* clustering technique for each airport, 2 clusters are imposed and formed based on the flights' average speed. The reason for choosing 2 clusters is that, if the flights must be separated based on their speed, the two groups we would want to form are high speed and low speed flights that correspond to commercial aviation and general aviation. The difference in speeds should be considerable.

But not all airports will handle both kinds of flights (or, at least, not in significant amounts). So speed clustering may not always be necessary. To identify whether or not it is necessary we must measure the quality of the resulting clusters to determine if they are adequate or not. There are several indices that can be used to indicate the quality of a clustering alternative. Some of the most popular are the Davis-Bouldin index, the Dunn index and the Silhouette coefficient. The Dunn index aims to identify dense and well separated clusters, which is not always the case for the studied route scenarios. The Davis-Bouldin index is based on intra-cluster and inter-cluster separations calculated for each cluster, and therefore gives the same weight to all clusters regardless of the number of datapoints that each of them contains. The Silhouette coefficient is calculated for each datapoint and the set of

Silhouette coefficients can be used as desired: a first averaging for each cluster and then an averaging at the cluster level would yield a very similar index to the Davis-Bouldin, whereas an overall averaging should give more importance to how adequate are the denser clusters. However, preliminary tests of both approaches have yielded very similar results, although further analysis should be done in the future. The method of choice will be the Silhouette coefficient (and ultimately its average), which is introduced next.

6.2.1 Silhouette Coefficient

"Silhouettes" were introduced by Rousseeuw in 1987 as a general graphical aid for interpretation and validation of cluster analysis [8]. In a Silhouettes calculation, the distance from each data point in a cluster to all other data points within the same cluster and to all data points in the closest cluster are determined. Thus Silhouettes provides a measure of how well a data point was classified when it was assigned to a cluster by according to both the tightness of the clusters and the separation between them.

The calculation of the Silhouette score for each datapoint is as follows,

$$S(i) = \frac{min\left(D_b(i,k)\right) - D_w(i)}{max\left[min\left(D_b(i,k)\right), D_w(i)\right]} \tag{6.2}$$

with $D_b(i,k)$ being the average distance from datapoint i to all other datapoints in another cluster k, thus $min\left(D_b(i,k)\right)$ is the $D_b(i,k)$ corresponding to the closest neighboring cluster, and $D_w(i)$ being the average distance from datapoint i to all other datapoints within the same cluster.

Eq. (6.2) can also be expressed as

$$S(i) = \begin{cases} 1 - \frac{D_w(i)}{min(D_b(i,k))} & \text{if } D_w(i) < min\,(D_b(i,k)) \\ 0 & \text{if } D_w(i) = min\,(D_b(i,k)) \\ 1 - \frac{min(D_b(i,k))}{D_w(i)} & \text{if } D_w(i) > min\,(D_b(i,k)) \end{cases} \tag{6.3}$$

Eq. (6.3) can better help understand the meaning of the obtained Silhouette values. A Silhouette close to 1.0 is obtained when the average distance from a datapoint to the other datapoints within its own cluster is smaller than the average distances to all data points in the closest cluster. A Silhouette close to zero indicates that the datapoint could equally well have been assigned to the neighbouring cluster. A negative Silhouette is obtained when the cluster assignment has been arbitrary, and the datapoint is actually closer to the neighboring cluster than to the other data points within its own cluster [9].

In this case, each datapoint (route) will be characterized by its average speed, and therefore the distance between two datapoints will be calculated as the difference between speeds, obtaining our dissimilarity measure. Once the 2 imposed clusters have been obtained and $S(i)$ has been calculated for each route, one can calculate the average Silhouette value for the speed clustering, $\bar{S}_S$,

$$\bar{S}_S = \frac{1}{N_r} \sum_{1}^{N_r} S(i) \tag{6.4}$$

where N_r is the number of routes being clustered. Following are two example cases for JAX and DAL airports.

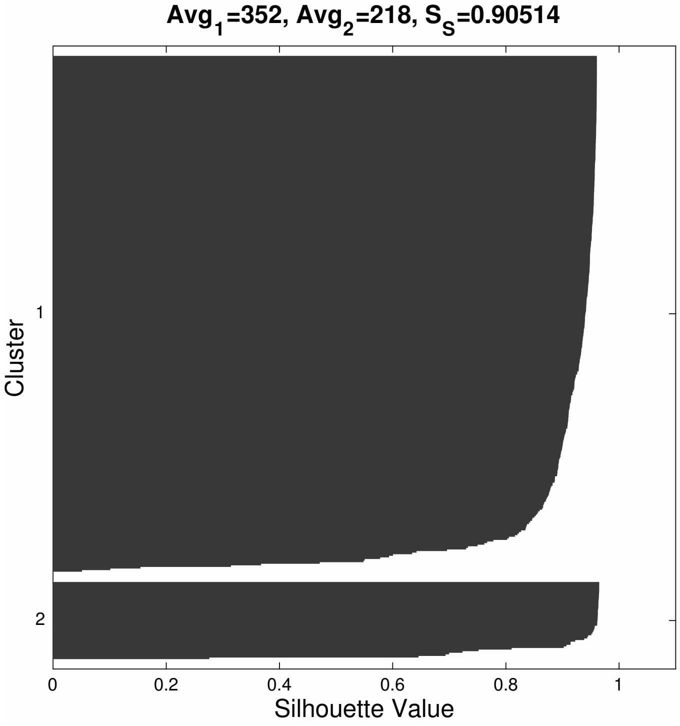

Figure 6.1: Silhouette scores for speed clustering, JAX.

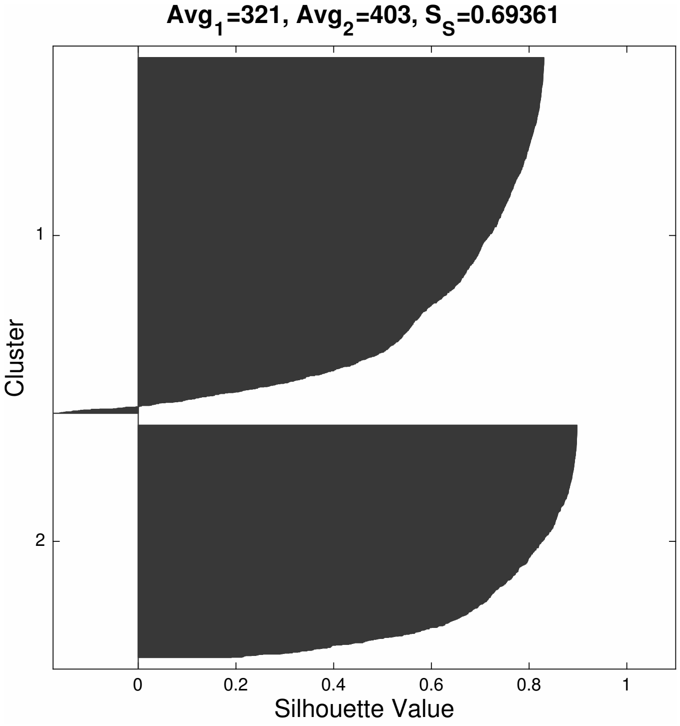

Figure 6.2: Silhouette scores for speed clustering, DAL.

In each of the figures we can see the routes being grouped in clusters, and the $S(i)$ value for each route is represented by a horizontal line, the length of which is $S(i)$. In addition, the average of the average speeds of the routes in each cluster are shown (values Avg_1 and Avg_2), as well as $\bar{S}_S$. As proposed by the original author in [8], $\bar{S}_S$ is a measure of how adequate (or natural) is a given cluster distribution. In our case a value of $\bar{S}_S$ close to 1 means that speed clustering in 2 clusters is very adequate, and negative values or values closer to 0 mean that it is not adequate. Only the scenario with 2 clusters is studied because of physical meaning. There may well be some situation where more than 2 clusters would be an adequate choice in terms of $\bar{S}_S$, but we don't want to complicate the speed classification in excess, and we limit it to the two defined groups. Only if two clear clusters are identified will it be performed.

By looking at Figure 6.2, with $\bar{S}_S = 0.91$, we can see a consistency in the $S(i)$ values within each cluster, most of them being very close to 1. We can also see a considerable difference between the average speeds in each cluster. On the other hand, in Figure 6.1, with $\bar{S}_S = 0.69$, we can see a wide variety of $S(i)$ values within each cluster, including some negative values. This indicates that there are no clear 2 groups of speeds in that dataset. The average speeds of each cluster are also much closer.

For this purpose, we found that setting a threshold of $\bar{S}_S = 0.80$ was adequate. For this choice, all scenarios with $\bar{S}_S > 0.80$ will be clustered by speed and two subgroups will be formed, and all others scenarios will not be clustered by speed.

The $\bar{S}_S$ threshold can be modified, and other techniques can be used to decide if speed clustering is appropriate. For example, one could use the Dip Test (introduced later) to identify unimodal speed distributions, but the results provided by $\bar{S}_S$ have been satisfactory.

Another approach could be to perform the speed clustering when the difference between average speeds of the two clusters is higher than a set threshold, or when the smaller average speed is smaller than a set threshold (corresponding to typical general aviation speeds). This approach could be combined with an $\bar{S}_S$ threshold.

6.3 Operational Similarity

After classifying the flights based on their average speed (if necessary) we must guarantee operational similarity within flights in a same subset. This is done by classifying the flights based on their origin and destination airports. For all flights inside the same speed category, subsets of those flights sharing the same origin and destination airport are formed.

Operational similarity is necessary in order to generate coherent clusters. If two geometrically similar flights with similar speeds, but directed towards different neighboring airports, were clustered together, it would be impossible to decide the destination airport for the resulting aggregate route.

Once the flights have been further classified based on their O/D pair, we have obtained our final subsets. Now flights in these subsets have similar speed and are operationally similar, so they can be clustered based on geometric similarity. The geometric clustering is central to this project and will be developed in detail in following chapters.

Chapter 7

Clustering Alternatives

We can distinguish two main types of clustering techniques: Partitional and Hierarchical. Their definitions are as follows (extracted from [10]):

- **Partitional** : Given a database of objects, a partitional clustering algorithm constructs partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the sum of squared distance from the mean within each cluster.

  One of the issues with such algorithms is their high complexity, as some of them exhaustively enumerate all possible groupings and try to find the global optimum. Even for a small number of objects, the number of partitions is huge. That's why, common solutions start with an initial, usually random, partition and proceed with its refinement. A better practice would be to run the partitional algorithm for different sets of initial points (considered as representatives) and investigate whether all solutions lead to the same final partition.

  Partitional Clustering algorithms try to locally improve a certain criterion. First, they compute the values of the similarity or distance, they order the results, and pick the one that optimizes the criterion. Hence, the majority of them could be considered as greedy-like algorithms.

- **Hierarchical** : Hierarchical algorithms create a hierarchical decomposition of the objects. They are either agglomerative (bottom-up) or divisive (top-down):

  - Agglomerative algorithms start with each object being a separate cluster itself, and successively merge groups according to a distance measure. The clustering may stop when all objects are in a single group or at any other point the user wants. These methods generally follow a greedy-like bottom-up merging.

  - Divisive algorithms follow the opposite strategy. They start with one group of all objects and successively split groups into smaller ones, until each object falls in one cluster, or as desired. Divisive approaches divide the data objects in disjoint groups at every step, and follow the same pattern until all objects fall into a separate cluster. This is similar to the approach followed by divide-and-conquer algorithms.

  Most of the times, both hierarchical clustering approaches suffer from the fact that once a merge or a split is committed, it cannot be undone or refined.

Partitional methods like *k-means* can become increasingly slow for large datasets (or even for some small particular datasets), and their solutions may vary when the clustering is carried out several times for the same dataset. The complexity involved to obtain good results using these algorithms (iterations, choice of initial points or *cluster seeds*, etc.) is another drawback. Also, convergence to a local minimum (criterion optimization) may produce counterintuitive results for some partitional methods. Another key limitation of *k-means* is its cluster model; the concept is based on spherical clusters that are separable in a way so that the mean value converges towards the cluster center. The clusters are expected to be of similar size, so that the assignment to the nearest cluster center is the correct assignment. In our dataset, clusters often have significantly different sizes. Last, algorithms like *k-means* require Euclidean distance metrics, which is something difficult to achieve for our purposes (route clustering).

For out type of dataset, the method of choice has been agglomerative hierarchical clustering. The speed and relative simplicity of this method make it adequate for our case, and the fact that no iterations are required and the solution for a given dataset is always the same are also advantageous. In addition, Hierarchical clustering accepts any kind of dissimilarity metric, and the observations are not even used, only a dissimilarity matrix is required. Refinement of the merges done by the hierarchical clustering could be achieved by combining it with partitional methods, but the results obtained without this refinement are considered satisfactory. The choice of the appropriate linkage method (discussed later) must be determined, and several tests will be carried out.

Besides the presented two main categories, other methods exist, like density-based clustering, grid-based clustering, model-based clustering and categorical data clustering. These or other alternatives have not been studied, and that possibility is left open.

Chapter 8

Dissimilarity Matrix

The dissimilarity matrix is an array in which the value of each element (i, j) represents the dissimilarity measure (result of the dissimilarity metric) between elements i and j in the dataset. The more dissimilar two elements are, the bigger the value of their dissimilarity measure.

On the other hand, a similarity matrix uses a similarity metric to characterize how similar two elements are, with bigger values meaning more similar. For hierarchical clustering, the input must be a dissimilarity metric (commonly referred to as distance metric).

There are many available options to characterize the distance between two n-dimensional points in a dataset. Some common options are the Euclidean distance, Squared Euclidean distance or Manhattan distance, to name a few. The choice becomes more complicated when we want to quantify the dissimilarity between two routes, represented by polygonal curves.

Some available options are based on the area between the curves, others are based on the distance between points in the curves. Both approaches have some implicit problems: the area-related approaches may not deal appropriately with curves intersecting each other, and the distance approaches may not deal appropriately with curves that are only different in a very localized region and are extremely similar everywhere else. Because in our dataset it is more common to have curves intersecting than curves with very localized differences, we will choose a measure based on the distance between points: the Fréchet distance, which is widely used in the literature to characterize dissimilarity between curves.

The Fréchet distance was introduced by Maurice Fréchet in 1906 [11]. It can be understood intuitively as follows. A man is walking a dog on a leash. Given two curves, the man can move on one curve, the dog on the other; both may vary their speed without backtracking. The Fréchet distance is the length of the shortest leash that is sufficient for the man and the dog to traverse those two curves and remain connected by the leash at all times. Figure 8.1 shows an example of the physical meaning of this distance metric in a discrete scenario like ours, where the two curves are described by a set of points.

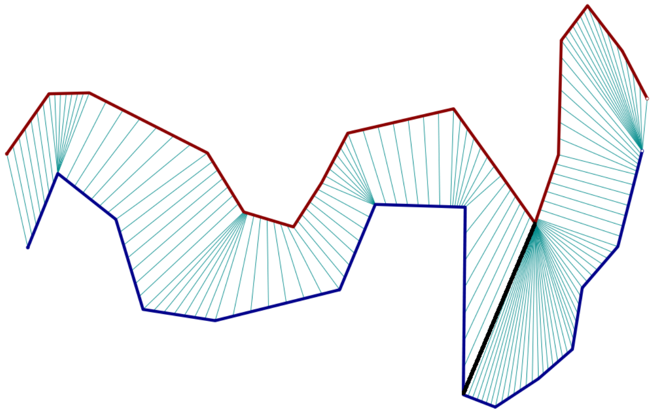

Figure 8.1: Example of discrete Fréchet distance.

The two routes are described by the thick red and blue polygonal curves. The thin lines in Figure 8.1 represent the connections between the positions of the man and dog at each time, and the thick line connecting the two curves is the longest of them: the Fréchet distance. There may be other ways to transit both curves that may result in the same 'shortest leash' length, but there is no way to transit them that results in a shorter leash allowing to connect both man and dog at all times.

The formal definition of the discrete Fréchet distance can be found in [12]. The algorithm used in our application to calculate the the discrete Fréchet distance is also that found in [12]. We present it in Algorithm 1.

Algorithm 1 Discrete Fréchet algorithm

```
function $dF(P, Q)$: real;
    input: polygonal curves $P = (u_1, ..., u_p)$ and $Q = (v_1, ..., v_q)$.
    return: $\delta_{dF}(P, Q)$
    $ca$: array $[1..p, 1..q]$ of real;
    function $c(i, j)$: real;
        if $ca(i, j) > -1$ then return $ca(i, j)$
        else if $i = 1$ and $j = 1$ then $ca(i, j) := d(u_1, v_1)$
        else if $i > 1$ and $j = 1$ then $ca(i, j) := max\{c(i-1, 1), d(u_i, v_1)\}$
        else if $i = 1$ and $j > 1$ then $ca(i, j) := max\{c(i, j-1), d(u_1, v_j)\}$
        else if $i > 1$ and $j > 1$ then $ca(i, j) := max\{min(c(i-1, j), c(i-1, j-1), c(i, j-1)), d(u_i, v_j)\}$
        else $ca(i, j) = \infty$
        end if
        return $ca(i, j)$;
    end function
    for $i = 1$ to $p$ do
        for $j = 1$ to $q$ do
            $ca(i, j) := -1.0$;
        end for
    end for
    return $c(p, q)$;
end function
```

It is important to note that

$$\delta_{dF}(P, Q) = 0 \Rightarrow P = Q \tag{8.1a}$$

$$\delta_{dF}(P, Q) \leq \delta_{dF}(P, R) + \delta_{dF}(R, Q) \tag{8.1b}$$

and therefore $\delta_{dF}(P, Q)$ defines a metric on the set of polygonal curves.

With the defined function, we will fill in our dissimilarity matrix, or Frechet Matrix (FM), such that

$$\begin{aligned} FM(i,j) &= dF(R_i, R_j) \\ &= \delta_{dF}(R_i, R_j) \end{aligned} \tag{8.2}$$

where R_i is the $i-th$ route and R_j is the $j-th$ route. Because

$$\delta_{dF}(R_i, R_j) = \delta_{dF}(R_j, R_i) \tag{8.3}$$

FM will be a symmetric matrix.

Chapter 9

Outlier Detection

Outlier data points are those points that are distant from other observations, or, in a broader sense, that do not conform to the rest of data. Detecting them is crucial to obtain quality clusters.

9.1 What are Outliers

Lets start by observing an example of outliers in a specific dataset to intuitively understand the meaning of outlier routes in our context.

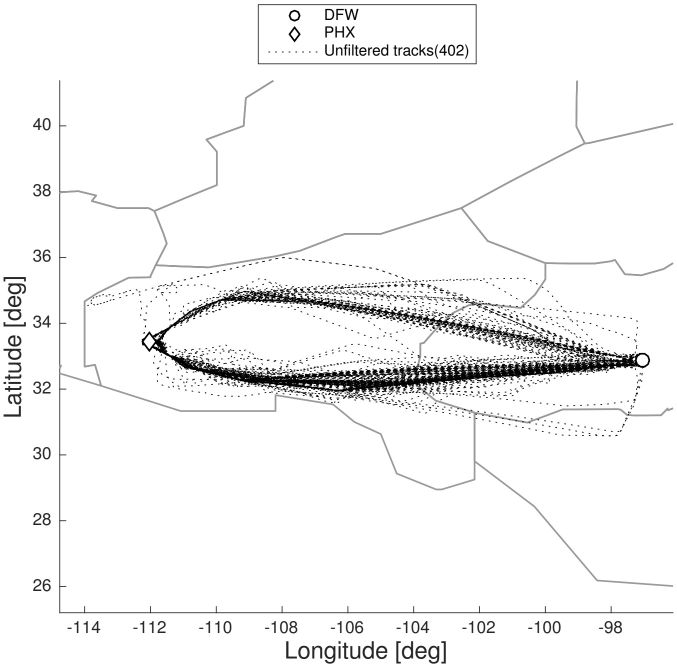

Figure 9.1: Unfiltered flights from DFW to PHX.

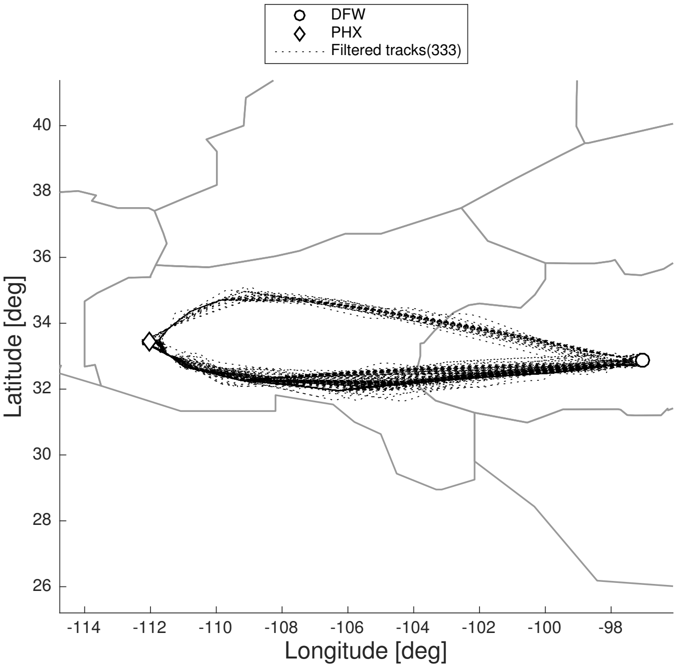

Figure 9.2: Filtered flights from DFW to PHX.

By looking at Figure 9.1 two very dense regions where most routes are concentrated can be identified. There are other routes that don't follow the dominant pattern but, if we were asked to identify the relevant flows in Figure 9.1, most of us would point out the two dense regions corresponding to the ones in Figure 9.10. The routes that have been eliminated in Figure 9.10 would be the outliers in this scenario.

9.2 Why Detect Outliers

Most clustering techniques have difficulty when the dataset contains outliers. Deleting them before clustering is a crucial step, and the following examples will show why.

Figure 9.3 corresponds to an unfiltered dataset, and Figure 9.4 and Figure 9.5 are the results of two different clustering techniques applied to the unfiltered dataset. We can see that both clustering techniques fail at recognizing the dominant patterns because of the presence of outliers.

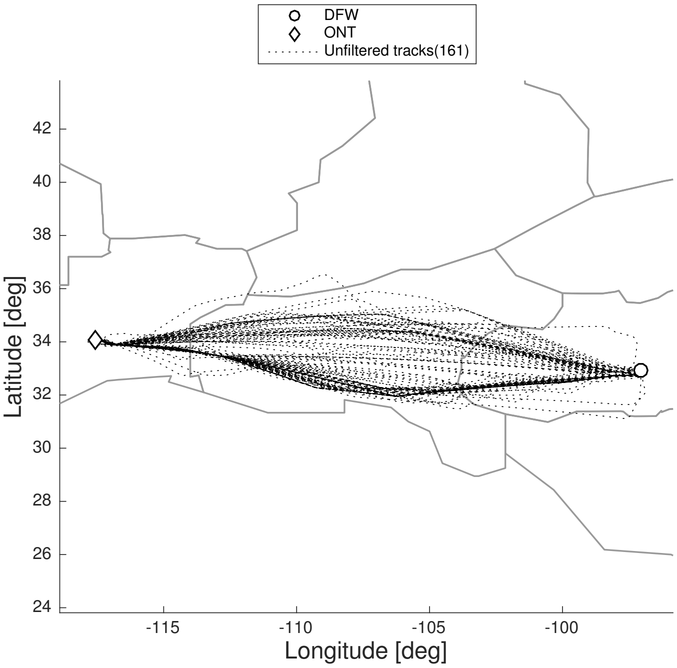

Figure 9.3: Unfiltered flights from DFW to ONT.

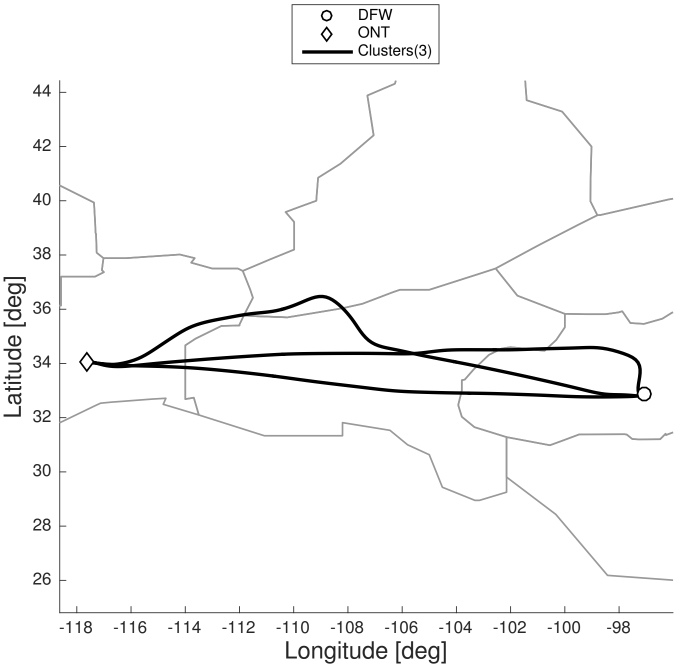

Figure 9.4: Incorrect clusters from DFW to ONT.

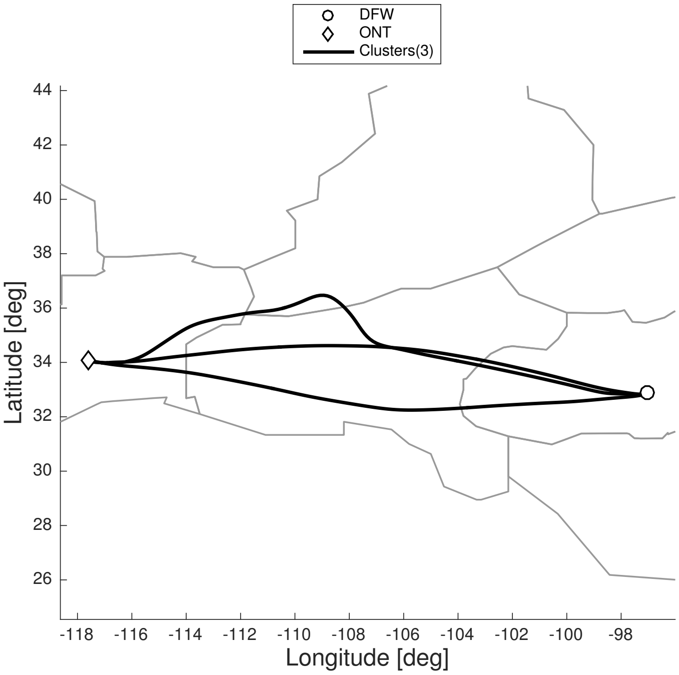

Figure 9.5: Incorrect clusters from DFW to ONT.

If, in stead, the dataset is filtered to detect the presence of outliers and eliminate them, the expected result is obtained.

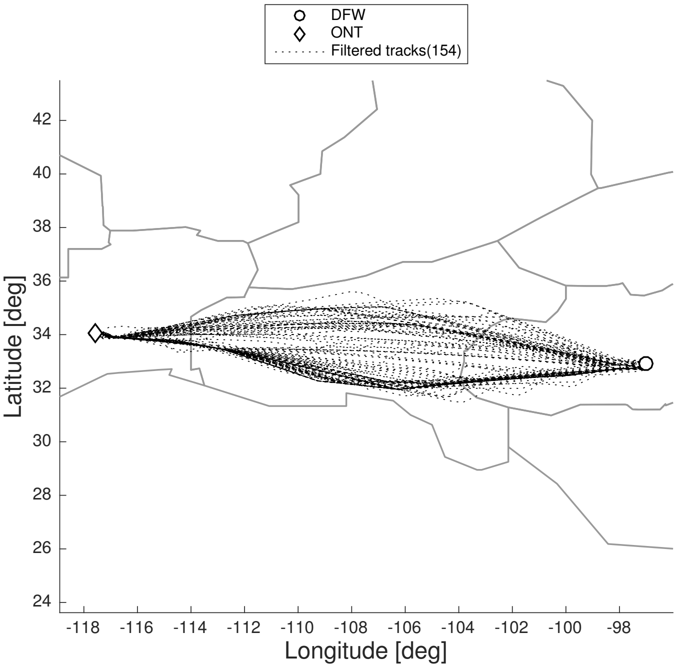

Figure 9.6: Filtered flights from DFW to ONT.

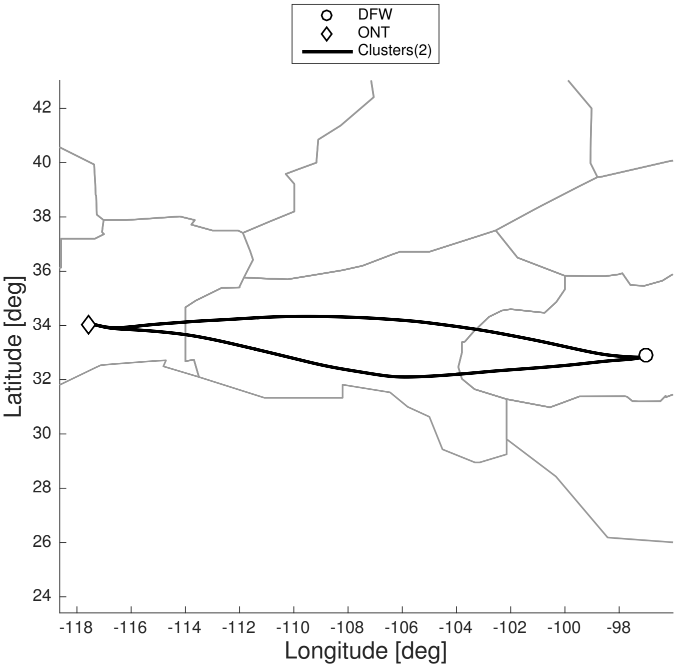

Figure 9.7: Correct clusters from DFW to ONT.

Figure 9.6 corresponds to the filtered dataset (without outliers), and Figure 9.7 corresponds to the result obtained applying any of the two clustering techniques applied in Figure 9.4 or Figure 9.5. We see that the result obtained is now physically meaningful and it represents the relevant flows in the dataset. Some tracks are contained somewhere in between the two main flows, and that information has been lost in the clustering process. The density of the main flows is much higher than that of the central routes and that's why they are not represented in the final clusters. This loss of information will always be present, specially in disperse clusters.

9.3 How to Detect Outliers

There are several algorithms available that are aimed at finding clusters of different sizes, shapes and densities in the presence of outliers. Some of them include ROCK [13], CURE [14], DBSCAN [15], CHAMELEON [16] and FAÇADE [17]. In addition, another method is developed in [18] and compared with the aforementioned methods. For a more detailed explanation of these methods and a comparison with our method of choice, reading of [18] is encouraged.

An outlier detection algorithm and a method to determine the number of clusters are developed in [18]. In this section, only the outlier detection algorithm will be presented and discussed.

One of the main challenges of other outlier detection methods is the need for user input parameters. These parameters must be tuned for different applications and datasets. In our aim for a fully automatized approach, a robust outlier detection method is desired. As we will see later, the method of choice still requires one input parameter, but the method is much more flexible and can correctly filter outliers in a very wide variety of scenarios for one single value of the parameter.

The algorithm presented in [18] is based on identifying low connectivity zones, understanding by connectivity the number of nearest neighbors to each datapoint. Outliers can be viewed as objects located in low density zones, or objects with low connectivity in opposition to the higher connectivity in the intra-cluster region. Other density algorithms use a similar approach, but they require external parameters to define the size of the target and the lower limit for density [15].

An iterative process is proposed where some characteristics of the system are used. The internal parameters are established based on the average nearest-neighbor distance (first parameter) and the average connectivity for all objects (second parameter). The latter depends on the previous parameter. In a convergence process, these parameters are automatically adjusted each time an elimination process is carried out, until the characteristics of the system stabilize – there are no significant variations form object to object –. The detection algorithm (from [18]) is the following,

Algorithm 2 Outlier detection algorithm

```
 1: function $outliers(FM)$
 2:     input: distance metric ($FM$), $P$.
 3:     return: outliers.
 4:     for $R_k = \{4, 2\}$ do
 5:         j=1
 6:         while discarded $\neq 0$ do
 7:             $\bar{d}_j = \frac{1}{N_r(j)} \sum_{i=1}^{N_r(j)} min(FM(i, [1 : i-1, i+1 : end]))$
 8:             $R := R_k \bar{d}_j$
 9:             $\bar{c}_j(R) = \frac{1}{N_r(j)} \sum_{i=1}^{N_r(j)} c_i = \frac{1}{N_r(j)} \sum_{i=1}^{N_r(j)} count(FM(i, [1 : i-1, i+1 : end])) < R)$
10:             Discard objects $i$ if $c_i < P\bar{c}_j(R)$
11:             j=j+1
12:         end while
13:     end for
14: end function
```

Notice that, on line 7 of Algorithm 2, $N_r(j)$ is the number of non-discarded routes at iteration j.

The outlier detection process goes as follows. First, $R_k = 4$. Then, the average nearest neighbor distance $\bar{d}_j$ is calculated using the routes that have not been discarded so far; for each route, the Fréchet distance to its nearest neighbor is found, and the average of all nearest-neighbor distances is calculated and assigned to $\bar{d}_j$. Then, the size of the target (or target radius) is calculated as $R := R_k \bar{d}_j$. Now the connectivity of each route c_i is defined as the number of routes that are closer than R to route i, and the average connectivity $\bar{c}_j(R)$ is calculated. Notice that the average connectivity is a function of the target radius R, because c_i is. Last, the routes with $c_i < P\bar{c}_j(R)$ are considered outliers and discarded. This process is repeated until no more outliers are found, and then it is all repeated for $R_k = 2$, a smaller target that results in a finer filtering.

It is important to emphasize that the value of the multiplier R_k to define the target size is not critical. As this algorithm relies on an overall comparison of the connectivity values of all objects, similar results are obtained for a wide range of multipliers. Of course, one should use big enough values for the multiplier so that the connectivity is higher than zero for most points in the dataset, implying that $R_k > 1$. On the other hand, very big values for R_k result in large overlapping regions and don't characterize the system appropriately. Therefore, low values for R_k, although larger than unity, are appropriate in general.

There is another multiplier, P, that we have specified as input in line 2. In the original algorithm P takes the value of 1/3, which is not adequate for our purposes as we will later see. The value of P must be determined by trial and error, but the fact it is a multiplier to $\bar{c}_j$ allows for one single P value to provide good results for a wide variety of scenarios.

Therefore, P must not be changed for each different filtering scenario, but rather be fixed based on a few example subsets and then be applied to the rest of subsets. Remember that a study of ASPM77 airports in the national airspace would yield 5,852 subsets: P can be fixed by choosing a few different subsets of different characteristics and studying its behavior, and then be applied to the rest of subsets or future study cases of the same nature (aircraft routes in the national airspace).

9.3.1 Choosing P

The parameter P is used to set the connectivity threshold that will separate outliers from routes that belong to clusters. Routes i with $c_i < P\bar{c_j}(R)$ are discarded, therefore, the higher the value of P, the higher the number of discarded routes. P can be understood as the "minimum connectivity ratio" acceptable for a route to be considered relevant, as

$$c_{lim} = P\bar{c_j}(R) \tag{9.1a}$$

$$P = \frac{c_{lim}}{\bar{c_j}(R)} \tag{9.1b}$$

with c_{lim} being the threshold connectivity value.

Let us introduce the two study cases that will be used in the following examples.

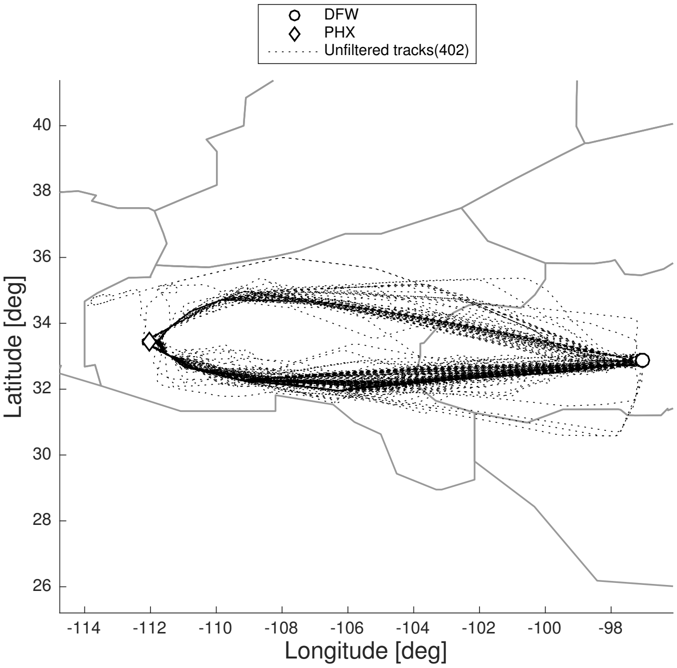

Figure 9.8: Unfiltered tracks to Phoenix (PHX).

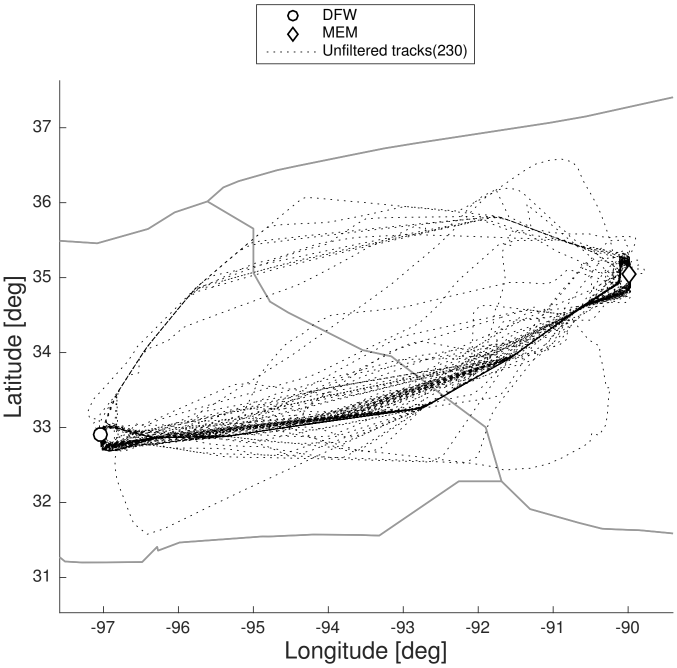

Figure 9.9: Unfiltered tracks to Memphis (MEM).

In Figure 9.8 we can see a scenario with two natural clusters of similarly high density, relatively compact, and some outliers. The scenario in Figure 9.9 consists of two clusters as well, but now the density variation between them is considerable. The presence of outliers is evident in both cases.

Effects of high P (P=0.2)

Lets see the effects of applying Algorithm 2 with $P = 0.2$ to the presented datasets.

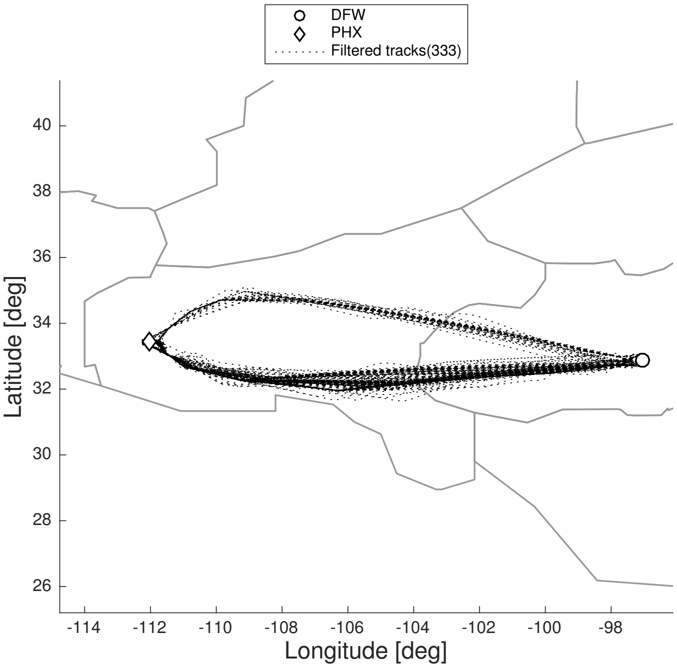

Figure 9.10: Filtered tracks to PHX, P=0.2.

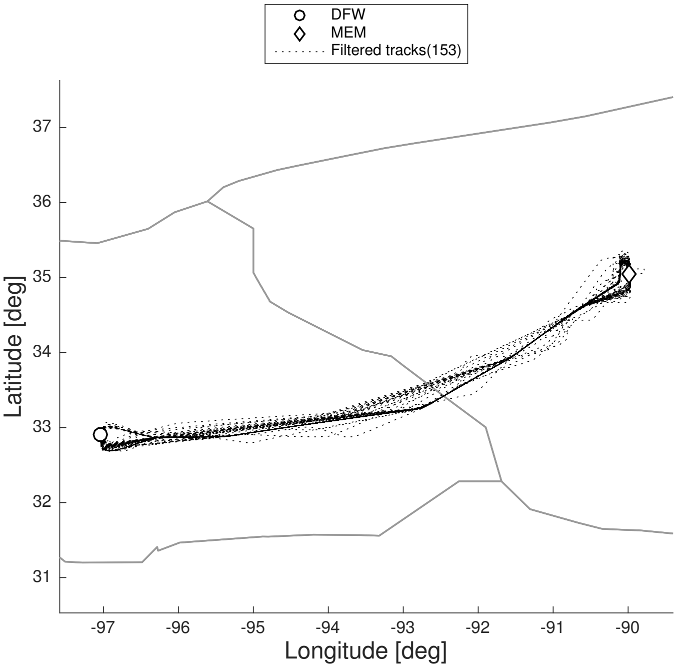

Figure 9.11: Filtered tracks to MEM, P=0.2.

In the case of PHX, where the clusters have similar density, the outliers are correctly detected.

But on the second case, from DFW to MEM, the low density cluster is incorrectly eliminated. We want to prevent the loss of relevant information, so the value of P will have to be adjusted.

The failure for high P values in cases with large density variations between clusters is due to the large number of high connectivity values c_i caused by the dense clusters, which raise the average value $\bar{c}_j$, and the routes in the less dense clusters end up having a connectivity $c_i < P\bar{c}_j(R)$. It is not impossible to deal with density variations, but P has to be adjusted.

Effects of low P (P=0.03)

Following are the results of applying Algorithm 2 using $P = 0.03$.

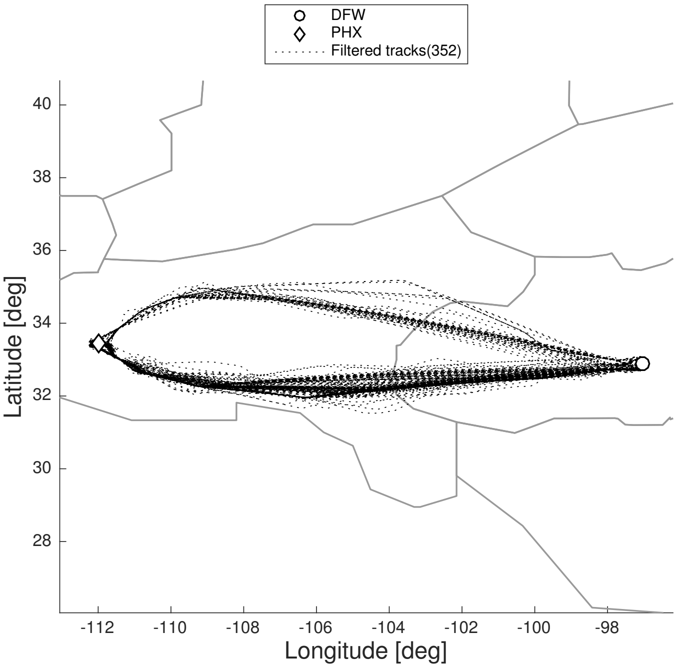

Figure 9.12: Filtered tracks to PHX, P=0.03.

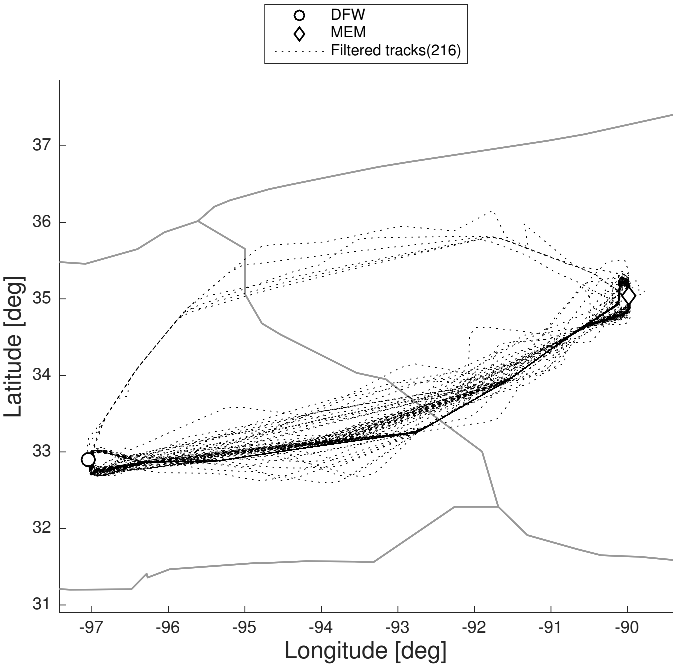

Figure 9.13: Filtered tracks to MEM, P=0.03.

In this case we obtain the expected results. It is important to notice that the filtered results for PHX in Figure 9.12 have not changed much with respect to Figure 9.10, but the results for MEM now include the less dense cluster – and also some subtle outliers around the dense cluster –. The reason behind this is that outliers are detected comparing their connectivity to the average connectivity. Thus for situations like Figure 9.8 where all clusters have high (and similar) densities there is a wide range of values of P for which we obtain correct results, because the difference between the connectivity of the outliers and the connectivity of the clustered routes is very large. On the other hand, clusters of low density are sensitive to the values of P. Thus, low values of P will provide satisfactory results in both situations.

Conclusions

High values of P behave correctly in the presence of dense clusters of similar density. When there are considerable cluster density variations, high P values may eliminate the less dense clusters. In all cases, severe outliers and subtle outliers are eliminated.

The choice of a low P value is necessary in order to not eliminate relevant information when there are big differences in cluster density. Low P values will still behave good in the presence of dense clusters of similar density, eliminating most outliers but occasionally not eliminating small structures of sub-clusters that may not be relevant – this structures will not affect the resulting number of clusters –. When the clusters have large density variations, low P values will not eliminate the less dense clusters, and some outliers (not severe) will be left around the main clusters. Therefore the clustering process will still be challenging because of the left outliers.

After some iterations using different values of P, $P = 0.03$ is chosen for all examples, and has provided satisfactory results in very different situations. There is flexibility in choosing the value of P and other values close to 0.03 will provide very similar results.

9.4 What to Do With Outliers

There are several possibilities about what to do with the detected outliers. In some situations it may be necessary to reassign the outliers to some of the formed clusters, if found appropriate. In our case this step is not needed as we are only interested in the end result (the aggregate routes) and there is no need to assign all routes to some cluster – we can simply eliminate them –. The reassignment would not substantially change the final shape of the clusters, as the datasets contain enough routes. In very small datasets this step could be beneficial.

Another option would be to look for other clusters in the outliers. This option would be necessary if high P values were used, because relevant information could be contained in the outliers. However, the process of reclustering outliers is not simple, as usually the same clusters that were obtained in the first clustering step will be obtained in subsequent steps, until some new clusters are formed. By the time new clusters are formed, that information may or may not be relevant.

Our approach is to use low P values to ensure that no relevant information is contained in the outliers, and eliminate them. The fact that some outliers are still left will be dealt with in the following steps.

Chapter 10

Finding the Number of Clusters

10.1 Linkage and Dendrogram

It is our goal now to introduce the concepts of *linkage* and *dendrogram* used in hierarchical clustering. They are closely related and will be presented simultaneously.

Agglomerative hierarchical clustering starts with each element in a separate cluster and then combines the clusters sequentially, reducing the number of clusters at each step until all objects belong to only one cluster. The first connection is necessarily between the two most similar elements, and it is then necessary to define the similarity between the newly formed group and the remaining elements. The way in which this similarity is measured is the *linkage* method. It is important to remark that for a given similarity metric of choice, each linkage method will result in different similarity measures between groups.

The *dendrogram* is a way to visualize the linkage distances (cophenetic distances) between the groups or elements in a dataset. In a dendrogram plot the x axis contains one entry for each element in the dataset (route IDs in this case), and each pair of groups is merged at a specific height equivalent to their linkage distance. A group can contain one or multiple elements. The result of a hierarchical clustering algorithm can be visualized in a dendrogram plot.

Following are some examples using a very simple set of routes that will serve to visualize the effects of some popular linkage algorithms, and the resulting dendrograms. The example set of routes is given in Figure 10.1

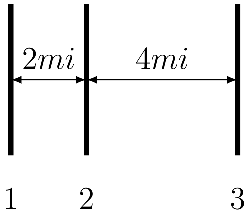

Figure 10.1: Example route distribution.

In this example case, routes 1 and 2 will always be linked first because they are the closest pair of elements in the dataset, and their cophenetic distance t (distance at which they are linked) will always be equal to their pairwise dissimilarity measure. On the other hand, the cophenetic distance between the group $\{1,2\}$ and route 3 will depend on the linkage method of choice. The dissimilarity metric we are using is the Fréchet distance, which for parallel routes is equivalent to their separation.

Single Linkage

For two groups of observations A and B, the single linkage cophenetic distance $t(A,B)$ is defined as

$$t(A,B) = min\{d(a,b){:}a{\in}A, b{\in}B\} \tag{10.1}$$

where $d(a,b)$ is the dissimilarity measure of choice between elements a and b. In this example case, the single linkage cophenetic distance between route 3 and group $\{1,2\}$ is equivalent to the distance between route 3 and route 2, $4mi$.

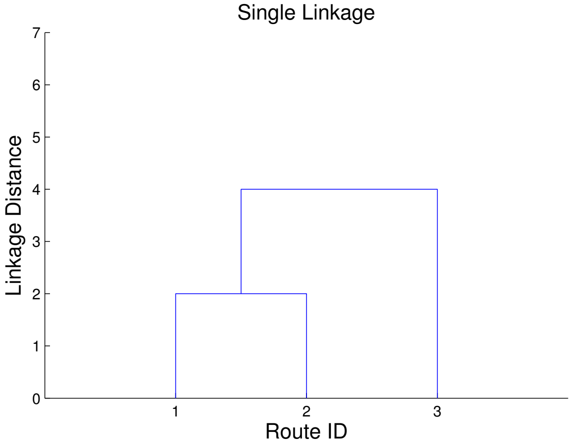

Figure 10.2: Single linkage dendrogram.

As we can see in the dendrogram (Figure 10.2) routes 1 and 2 are linked at $2mi$, and this group is linked with route 3 at $4mi$.

Complete Linkage

For two groups of observations A and B, the complete linkage cophenetic distance $t(A, B)$ is defined as

$$t(A, B) = max\{d(a, b){:}a{\in}A, b{\in}B\} \tag{10.2}$$

where $d(a, b)$ is the dissimilarity measure of choice between elements a and b. In this example case, the complete linkage cophenetic distance between route 3 and group $\{1, 2\}$ is equivalent to the distance between route 3 and route 1, $6mi$.

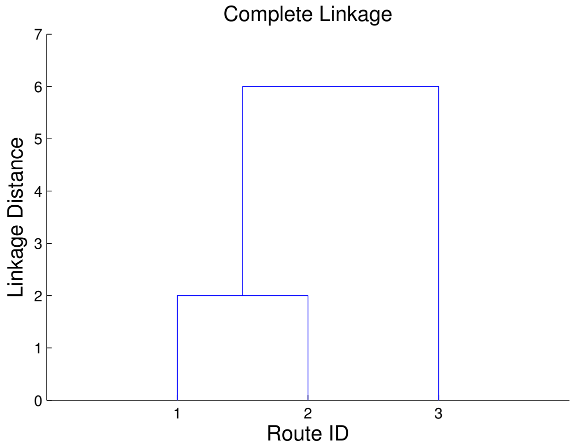

Figure 10.3: Complete linkage dendrogram.

As we can see in the dendrogram (Figure 10.3) routes 1 and 2 are linked at $2mi$, and this group is linked with route 3 at $6mi$.

Average Linkage

For two groups of observations A and B, the average linkage cophenetic distance $t(A, B)$ is defined as

$$t(A, B) = \frac{1}{|A|\,|B|} \sum_{a \in A} \sum_{b \in B} d(a, b) \tag{10.3}$$

where $d(a, b)$ is the dissimilarity measure of choice between elements a and b. In this example case, the average linkage cophenetic distance between route 3 and group $\{1, 2\}$ is equivalent to the average distance between route 3 and route 2, and between route 3 and route 1, $5mi$.

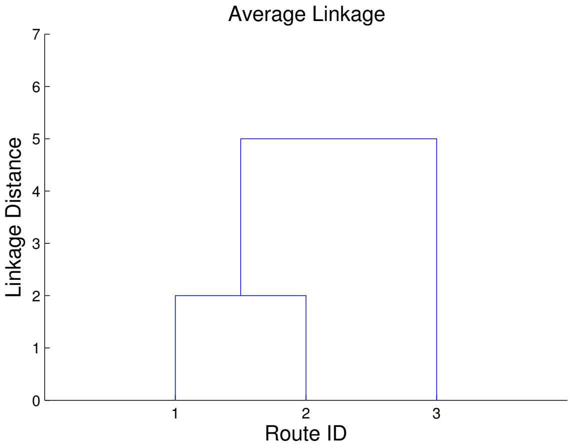

Figure 10.4: Average linkage dendrogram.

As we can see in the dendrogram (Figure 10.4) routes 1 and 2 are linked at $2mi$, and this group is linked with route 3 at $5mi$.

Linkage Methods Comparison

Besides the three presented linkage methods there are other alternatives. Some of the other popular alternatives include *centroid* method and *Ward's* method. These alternatives require the use of en Euclidean distance metric and therefore cannot be used with the Fréchet distance. We will see that their features are very similar to the *average* linkage.

Following is a table summarizing the advantages and disadvantages of the five introduced linkage methods.

Linkage	**Advantages**	**Disadvantages**
Single	Shape & size	Outliers & density variation
Complete	Outliers	Can break large clusters
Average	Outliers	Shape & size
Centroid	Outliers	Shape & size
Ward's	Outliers	Shape & size

Table 10.1: Linkage methods comparison.

It is important to notice that single linkage can provide incorrect results for scenarios with outliers and density variations. These are precisely the two main problems of our dataset so one could advance that this linkage method would not be the most suitable for route clustering.

Pruning the Dendrogram

We have introduced the dendrogram as a graphic representation of the linkage distances in a hierarchical clustering process. The dendrogram is also used to visualize the formation of clusters.

Once a dendrogram is obtained for a specific scenario, the dendrogram can be pruned at a certain height (cophenetic distance): the number of 'branches' that are cut by a horizontal line placed at the specified height determines the number of resulting clusters, and all the elements that are contained in the lower levels of each branch will form that specific cluster.

The height (distance) at which the dendrogram is pruned represents the limit linkage distance that the user is willing to accept between two subclusters inside that cluster: i.e., no pair of subclusters inside that cluster will be more dissimilar than the height at which the dendrogram was pruned.

Let's take one of the presented dendrograms, corresponding to complete linkage for example, Figure 10.3. If the dendrogram is pruned at a height of between $0mi$ and $2mi$, 3 clusters will be obtained, one for each route. If it is pruned between $2mi$ and $6mi$, 2 clusters will be obtained, one for routes $\{1, 2\}$ and one for route 3. If the dendrogram is pruned at a height greater than $6mi$, all three routes will become one single cluster.

Determining the appropriate height to prune a dendrogram (or, equivalently, the appropriate number of clusters) is one of the biggest challenges in hierarchical clustering – and in any kind of clustering –. Several automatic approaches to determining the number of clusters have been developed [19], [20], [21], [22], [23], [24].

Natural clusters can be clearly identified (and the process easily automatized) when inter-cluster separations are significantly higher than intra-cluster separations (with homogeneous intra-cluster separations). The process becomes more complex when clusters are defined at different levels of resolution; some clusters are in turn formed by smaller sub-clusters, that can be regarded as an internal structure [18].

It is frequent in the exisiting methods to form clusters when the inter-group separations are distinct enough from the intra-group separations. This approach usually results in an excessive heterogeneity in inter-group division, and the methods tend to go into the fine structure of clusters. The result is usually an excessive division of data, and the identification of clusters at all resolution levels, identifying clusters and sub-clusters at the same time. Some examples of this approach can be found in [21], [22], [24], which are based on reachability plots produced by the *optics* algorithm [25], [26]. The reachibility plots combine selective linkage and density analysis.

The approach presented in [18] aims at identifying the groups from an external global view of all the system. This project intends to achieve similar results from the same conceptual point of view, adapted to the restrictions of our dataset. The main challenges will be the heterogeneity in inter-cluster and intra-cluster separation due to the differences in cluster density, the abundance of outliers and disperse clusters and the presence of internal structures or sub-clusters.

The simplest approach consisting of a fixed distance threshold will be tested and characterized, as well as the method developed in [18], which is conceptually similar to our objective. Last, the developed method will be tested and compared to the other two alternatives.

Cophenetic Coefficient

We have just presented various linkage techniques that will result in different linkage distances between groups, and thus in different dendograms for the same dataset. It is to be expected that some linkage methods may be more appropriate than others to faithfully represent specific datasets. The coefficient introduced here is a measure to quantify this effect.

The cophenetic distance between two observations is represented in a dendrogram by the height of the link at which those two observations are first joined. That height is the distance between the two subclusters that are merged by that link.

The cophenetic coefficient (or cophenetic correlation) [27] for a cluster tree is defined as the linear correlation coefficient between the cophenetic distances obtained from the tree, and the original distances (or dissimilarities) used to construct the tree. Thus, it is a measure of how faithfully the tree represents the dissimilarities among observations.

Define

$d(i, j)$= the dissimilarity measure between observations i and j, and $\bar{d}$ its average value.

$t(i,j)$= the cophenetic distance between observations i and j, and $\bar{t}$ its average value.

Then the cophenetic coefficient c is calculated as follows,

$$c = \frac{\sum_{i<j}(d(i,j)-\bar{d})(t(i,j)-\bar{t})}{\sqrt{[\sum_{i<j}(d(i,j)-\bar{d})^2][\sum_{i<j}(t(i,j)-\bar{t})^2]}} \tag{10.4}$$

Values of c close to 1 indicate a faithful representation of the dataset by the dendrogram, and values close to 0 indicate the opposite.

Effects of the Linkage in the Dendrogram

The following scenario consists on the filtered flights from DFW to SEA.

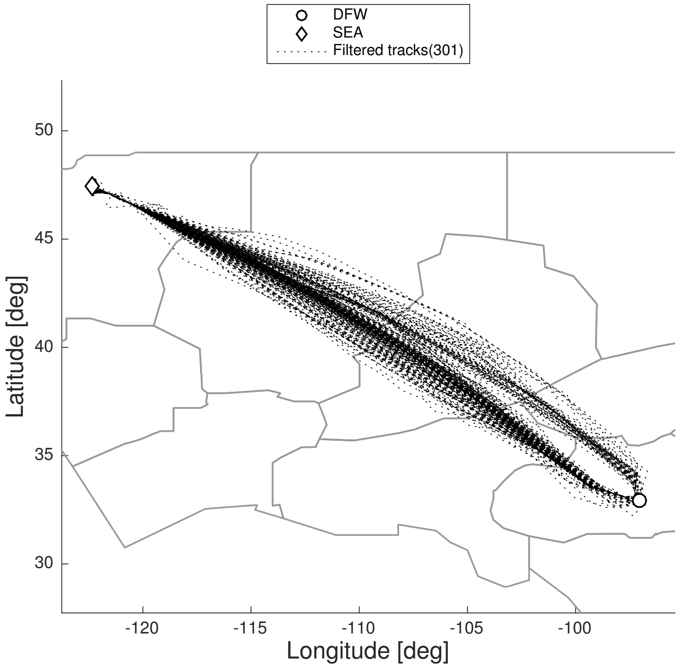

Figure 10.5: Filtered tracks to SEA.

An automatic method to determine the correct number of clusters has not been presented yet, so for now lets assume we impose 2 clusters for this distribution by visual analysis.

The presented dendrograms correspond to the three studied linkage methods, and horizontal dashed lines delimit the region where the dendrogram should be pruned in each case in order to obtain 2 clusters as a result.

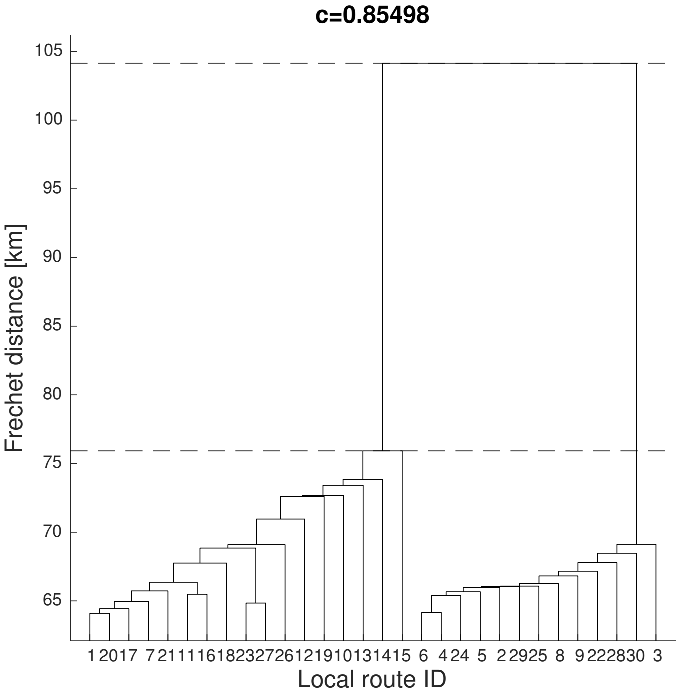

Figure 10.6: Dendrogram, single linkage, SEA.

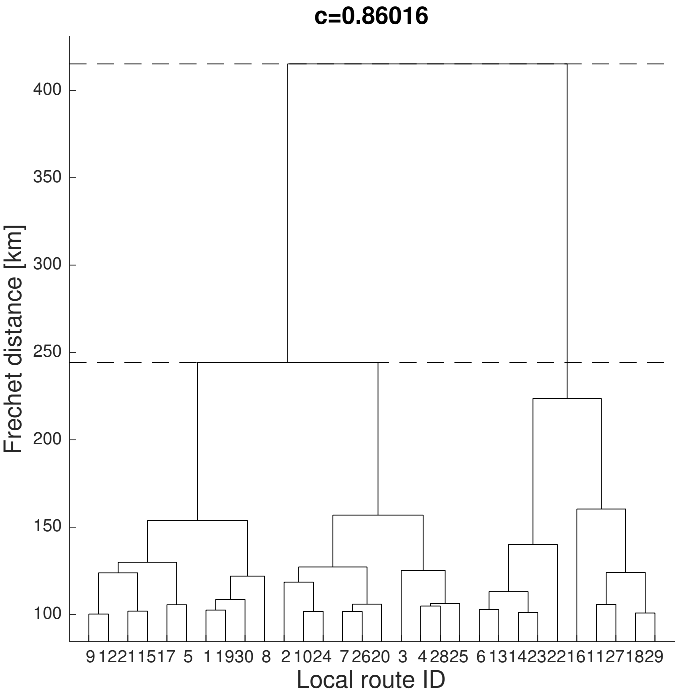

Figure 10.7: Dendrogram, complete linkage, SEA.

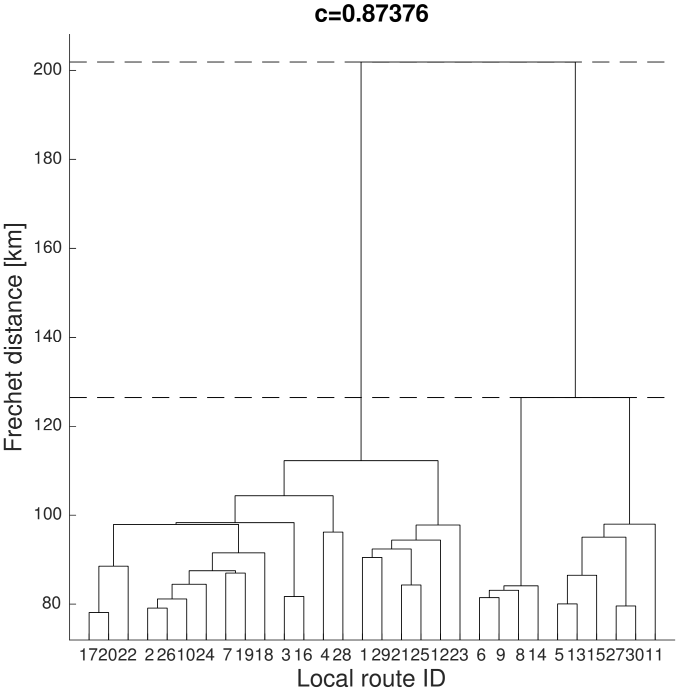

Figure 10.8: Dendrogram, average linkage, SEA.

For clarity, only a maximum of 30 groups are plotted in the x axis of the dendrograms. When there are more than 30 routes in the dataset, the groups in the x axis do not correspond to individual elements, but to the distribution corresponding to 30 clusters.

It is interesting to notice that there is no overlap whatsoever between the three ranges of linkage distances (Fréchet distances) that would result in two clusters – in fact, there isn't overlap between the range of any two linkage methods –. There is no common Frechét distance limit that would provide 2 clusters as a result for all three linkages. The relevance and influence of the linkage method becomes apparent.

In the following sections the three mentioned linkage alternatives (single, complete and average) will be tested, together with some methods to determine the number of clusters of a scenario.

10.2 Criteria for Performance Evaluation

In order to analyze the performance of the different methods to choose the number of clusters of a dataset, it is necessary to determine the desired results beforehand. Out of the 76 subsets that are contained in our dataset (no speed clustering is required, and there is one subset for each one of the 77 ASPM77 destinations excluding DFW itself), only 53 of them will be considered in this study case. Some of the excluded subsets don't contain enough flights to carry out clustering (a lower limit of 10 flights per subset has been imposed), and others don't have any clear recognizable patterns and therefore there is no *target* result to be expected and considered correct.

The 53 selected subsets have been visually analyzed to determine the expected number of clusters and their expected geometric appearance. The results of each tested linkage method and pruning technique have been compared with the expected results, and the success rate has been calculated as the proportion of expected results achieved.

With very few exceptions, when the number of clusters determined by a method is correct, their geometric distribution is, as well. Therefore, the number of clusters determined by a method will be used as measure.

Each presented pruning technique is tested for single, complete and average linkage, for the 53 selected subsets of the original 60-day dataset.

In all cases (for all linkage methods and pruning techniques) the used datasets are the same, and have been previously filtered using the outlier detection algorithm for $P = 0.03$.

10.3 Fixed Distance Threshold, D_t

A fixed distance threshold D_t is the most basic way to determine where to prune a dendrogram. The height where the dendrogram is pruned is previously fixed for all cases, independently of the subset.

In our case the distance threshold D_t corresponds to a Fréchet threshold F_t, as Fréchet is the used distance metric.

Because there is a wide range of values that F_t can take, the process to test its performance has been the following: for each subset, the range of F_t values (taken at unitary steps, in $[km]$) that would provide the expected number of clusters is found. This range is stored, and finally all the ranges of F_t are plotted in a histogram. For each value of F_t, the ratio of correct predictions vs. total cases is calculated and plotted in the vertical axis.

Next are the obtained histograms for all three linkage methods. The mode of each distribution, with its correct clustering ratio p, is specified in each plot.

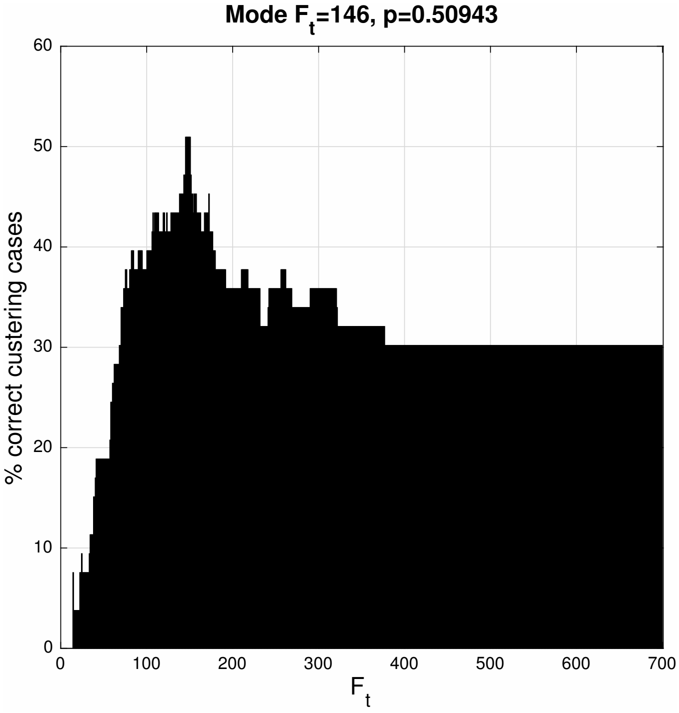

Figure 10.9: D_t histogram, single linkage.

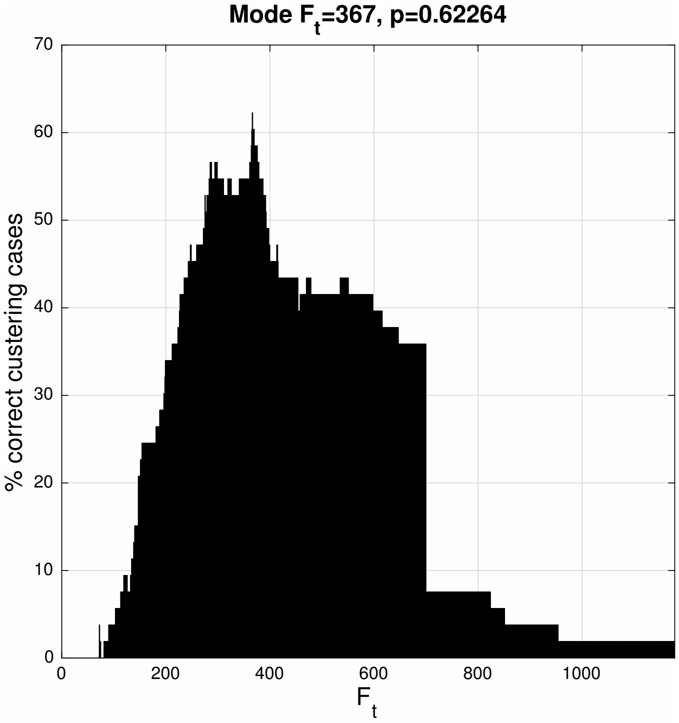

Figure 10.10: D_t histogram, complete linkage.

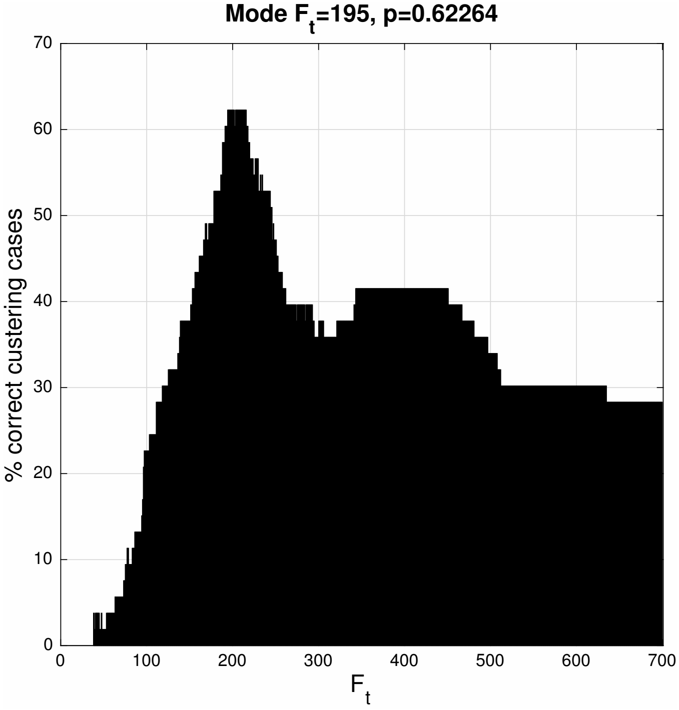

Figure 10.11: D_t histogram, average linkage.

As expected, the optimum F_t values for the different linkage methods differ considerably. Most importantly, the maximum achievable ratio of correct clustering cases is of 62%. This is a measure of the flexibility of the method, and proves it has trouble when clustering subsets of very different nature (different length scales, densities and shapes).

In addition, single linkage is the linkage method providing the worst performance, as anticipated.

It must be noted that the optimal F_t values are not known a priori, and determining these values for a scenario where the expected solution is unknown would be really difficult. Even when the expected solution is known, determining these values is an arduous task that requires individual analysis of all cases, which defeats the purpose of this work and should be avoided.

10.4 Descriptive Function, DF

An alternative and automatic method to determine the height to prune the dendrogram is proposed in [18]. This method is based on the squared single-linkage distances of the dataset and the obtained pruning height is dependent on each dataset. The method is presented as a flexible general approach. It is introduced as an appropriate method for single linkage, but it will be tested for all single, complete and average linkages.

The first step in this method is to calculate the descriptive function DF for each pair of objects. The order of the objects present in the final association vector is the following: when two objects are first associated, they are placed in consecutive positions. When an additional object or group is further associated into an existing structure (single element or group), it will appear in the vector before or after the original group. Therefore, formed blocks suffer no changes during the subsequent association steps.

The descriptive function for a pair of consecutive objects i, $i+1$ corresponds to the squared minimal distance measure of all linkage steps in which both objects participate (or cophenetic distance),

$$DF_{i,i+1} = t^2_{i,i+1} \tag{10.5}$$

The presented descriptive function will produce localized higher peaks corresponding to a high probability of inter-cluster separation, and low value regions indicating a high probability of intra-cluster association.

An inter-cluster separation DF_{sep} is found in the descriptive function as

$$DF_{sep} = K(Q_3 - Q_1) \tag{10.6}$$

where Q_3 and Q_1 are the upper limits of the first and third quartile of the distribution of values in the descriptive function.

In [18], the chosen value for the parameter K is $K = 6$. For datasets of routes with very compact clusters of similar densities, even in the presence of many outliers, the method presented in [18] is very successful, including the chosen values for the parameters $P = 1/3$ (outlier detection) and $K = 6$ (pruning the dendrogram). The use of these parameters is not appropriate in other complex scenarios with heavy density variations and not very compact clusters (as we have partially shown in *Subsection 9.3.1 Choosing P*). Next we will determine if there is any appropriate choice of the parameter K that provides satisfactory success rates, or whether this method is not flexible enough for our clustering scenarios.

The process followed to test this method is very similar to the one followed in *Section 10.3 Fixed Distance Threshold, D_t*. Now, for each subset, the range of values of K providing the expected number of clusters is found (at unitary steps) and finally all values are plotted in a histogram. Following are the obtained results.

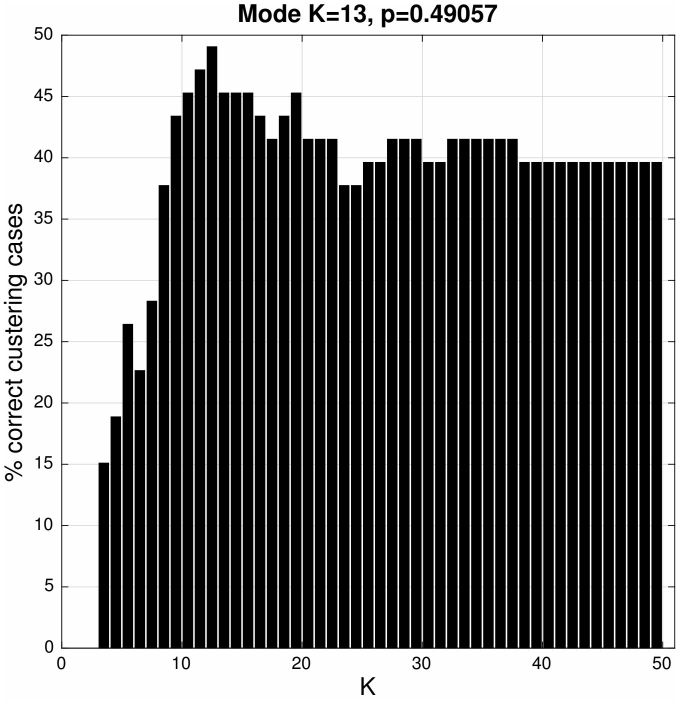

Figure 10.12: K histogram, single linkage.

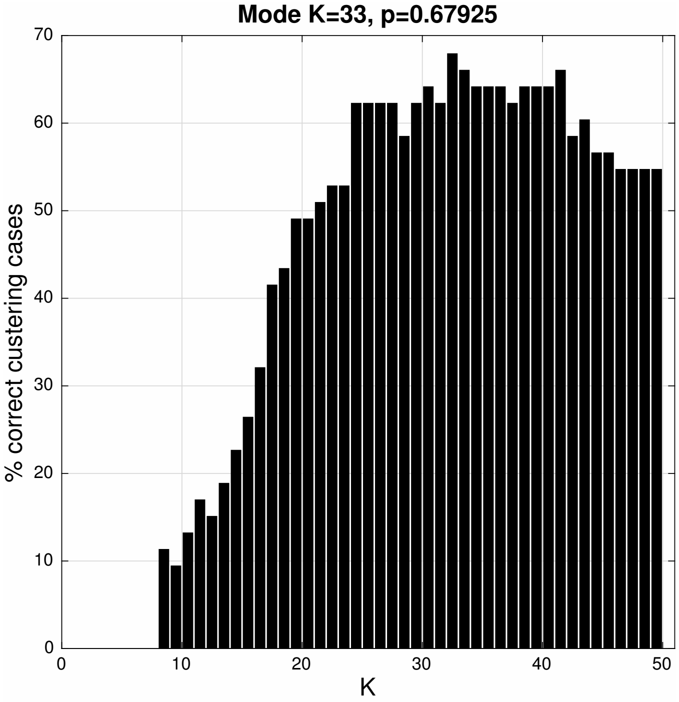

Figure 10.13: K histogram, complete linkage.

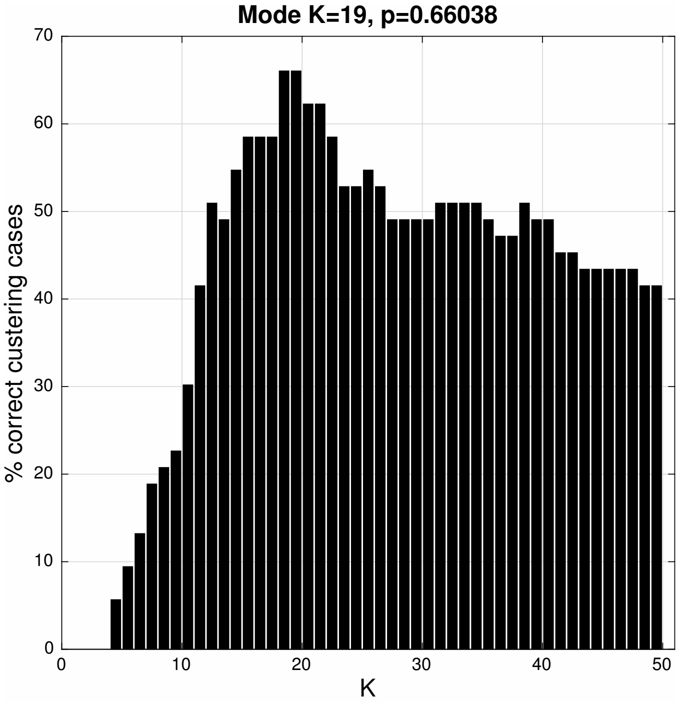

Figure 10.14: K histogram, average linkage.

The DF method performs better than the D_t method for complete and average linkages, obtaining maximum performances of 68% and 66% respectively. It performs slightly worse than D_t for single linkage.

The obtained results, if somewhat better, are very similar to those obtained for the D_t method. Again, single linkage provides the worst performance for our dataset, as expected. The optimal values for K also differ considerably between linkage methods, adding to the difficulty of determining them.

Last it must be noted that, again, the optimal K values are not known a priori, and determining them is equally challenging as determining the optimal F_t values.

10.5 Problem of the Presented Methods

As indicated by the success rates of the presented methods, they lack flexibility to deal with subsets of very different nature. This can be better visualized with an example.

For the following scenarios and after visual analysis the desired number of clusters is 2 for both cases.

Seattle-Tacoma International Airport (SEA)

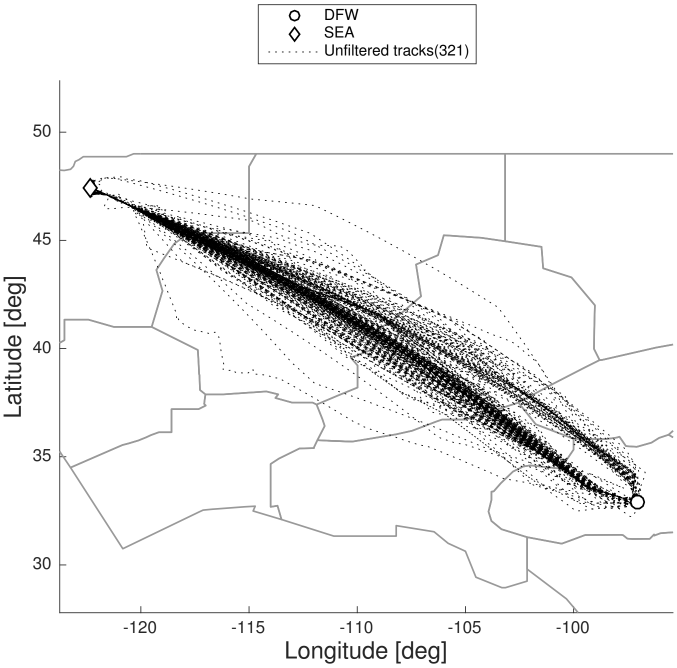

Figure 10.15: Unfiltered tracks, SEA.

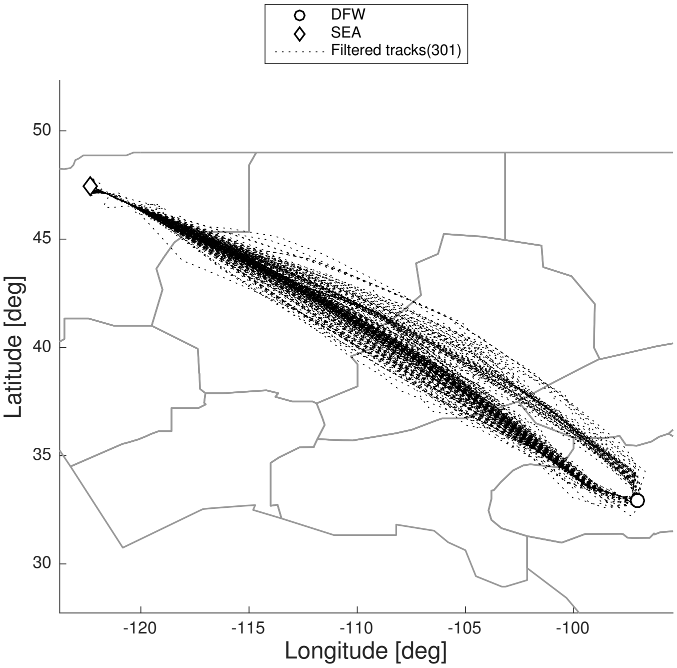

Figure 10.16: Filtered tracks, SEA.

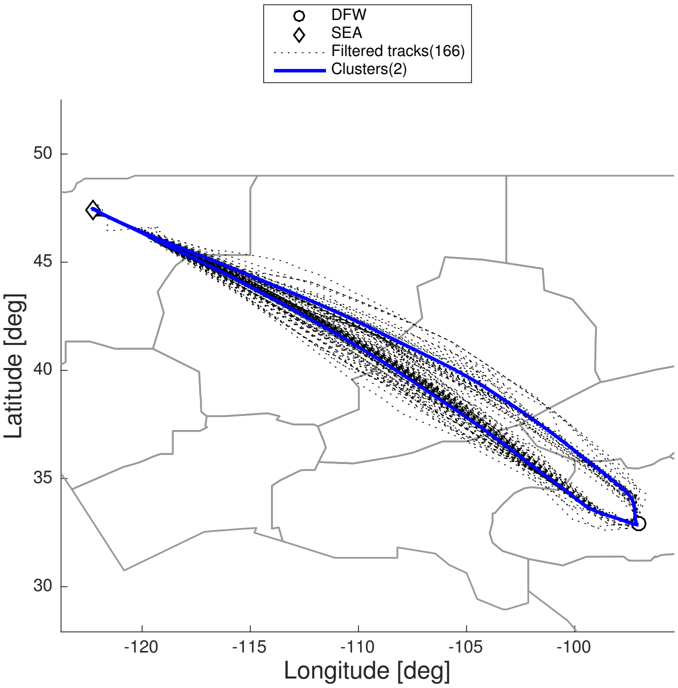

Figure 10.17: Tracks and clusters, SEA.

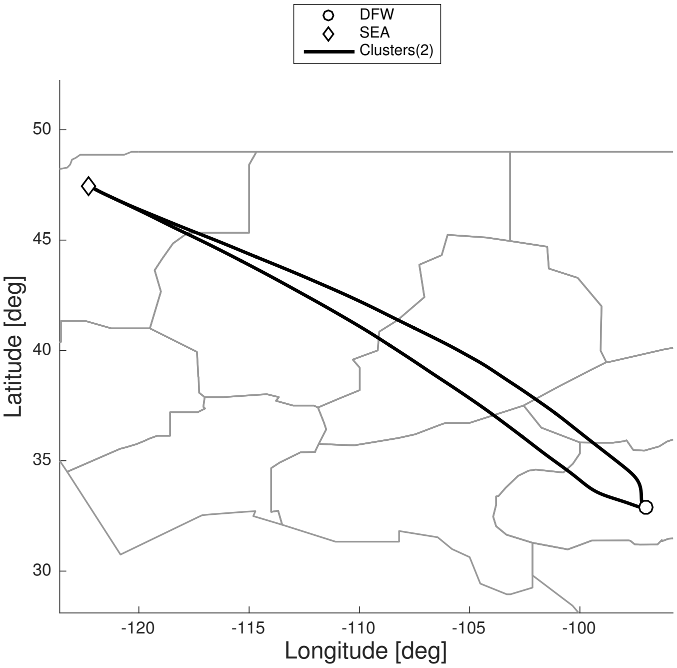

Figure 10.18: Clusters, SEA

Pittsburgh International Airport (PIT)

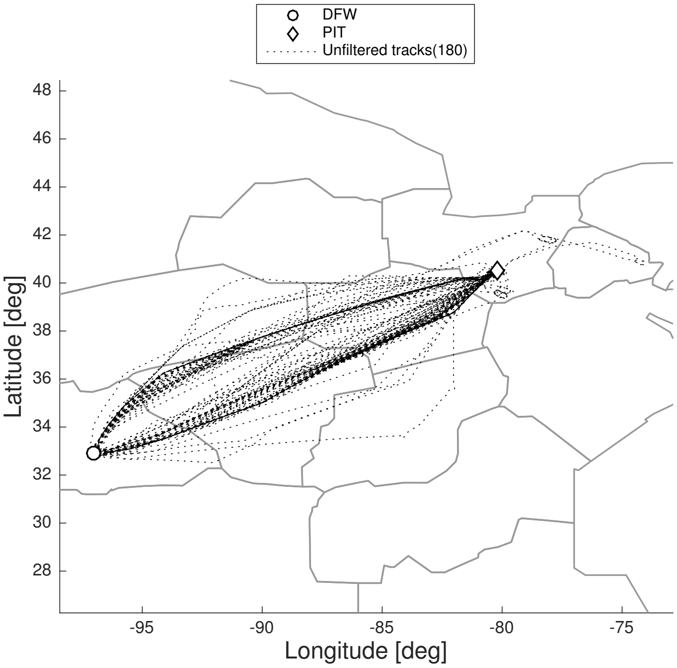

Figure 10.19: Unfiltered tracks, PIT.

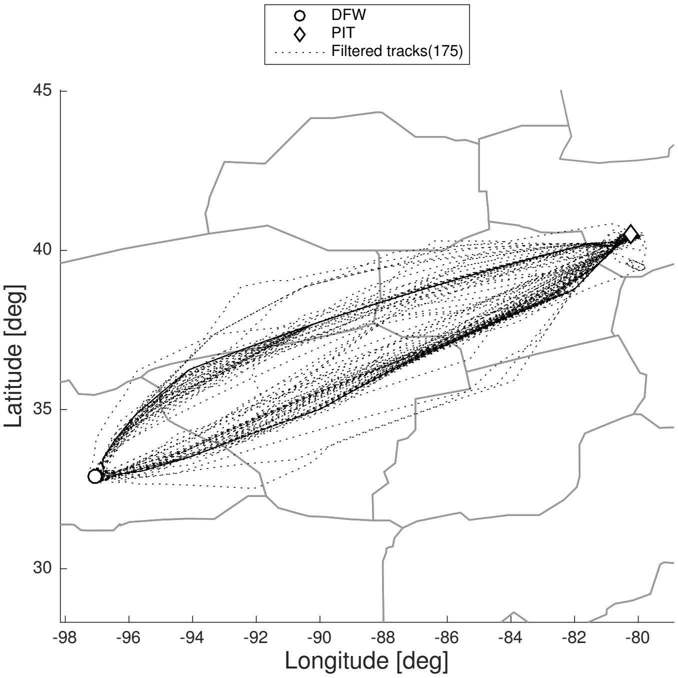

Figure 10.20: Filtered tracks, PIT.

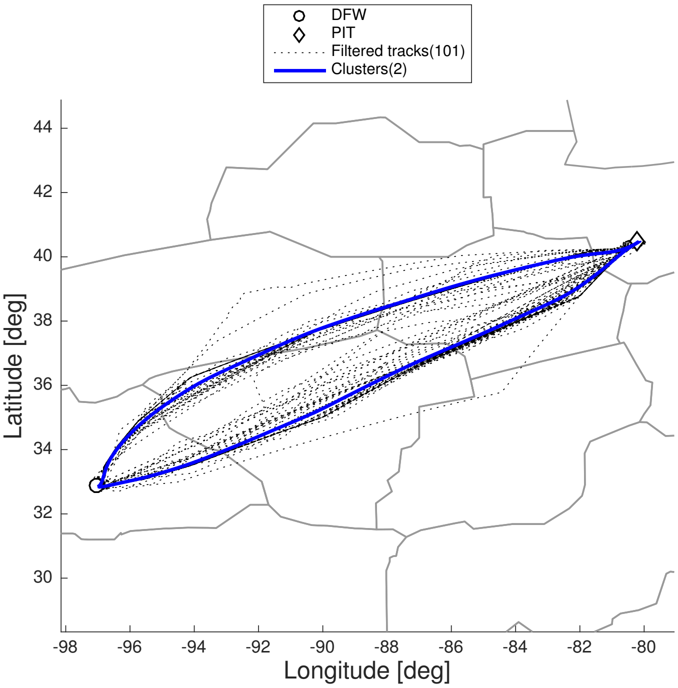

Figure 10.21: Tracks and clusters, PIT.

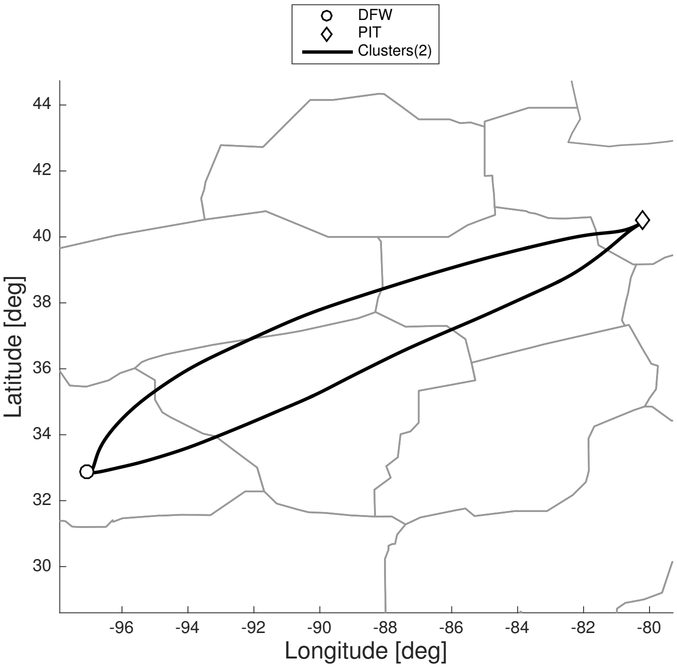

Figure 10.22: Clusters, PIT

If these results had to be achieved by pruning the dendrogram using the presented methods, the parameters to use should be contained in the following ranges,

	SEA	**PIT**
K range	[6 - 13]	[19 - 23]
$\boldsymbol{D_t}$ range [km]	[135 - 199]	[204 - 274]

Table 10.2: Valid ranges for parameters, SEA and PIT.

There is no overlap in the ranges of any parameter, so no matter what method was used, the desired result could only be obtained in one of the clustering cases. The presented scenario is an extreme case where both methods fail. It is common in our dataset to find pairs of subsets where at least one of the methods is unable to produce the expected results in both cases.

10.6 Average Silhouette, S_{max}

Using $\boldsymbol{S_{max}}$

Rousseeuw proposed in [8] the use of the average silhouette value of a clustering scenario as an indicator of how appropriate it was. We will apply this concept to determine where to prune the dendrogram for our geometric clustering.

Lets remember the definition of Silhouette score, Eq. (6.2),

$$S(i) = \frac{min\left(D_b(i,k)\right) - D_w(i)}{max\left[min\left(D_b(i,k)\right), D_w(i)\right]}$$

The dissimilarity between two routes will now correspond to their discrete Fréchet distance. Now $min(D_b(i,k))$ corresponds to the average Fréchet distance between route i and all the routes in the nearest neighboring cluster k, and $D_w(i)$ corresponds to the average Fréchet distance between route i and all other routes in its own cluster.

The average silhouette value for a subset, $\bar{S}_N$, will be calculated as the average of the silhouette values of all routes in that subset,

$$\bar{S}_N = \frac{1}{N_r} \sum_{1}^{N_r} S(i) \tag{10.7}$$

where N_r is the number of routes in the subset.

In order to visualize the meaning of $\bar{S}_N$, some scenarios for the same dataset are attached with different clustering alternatives, their corresponding aggregate routes, and the $\bar{S}_N$ values.

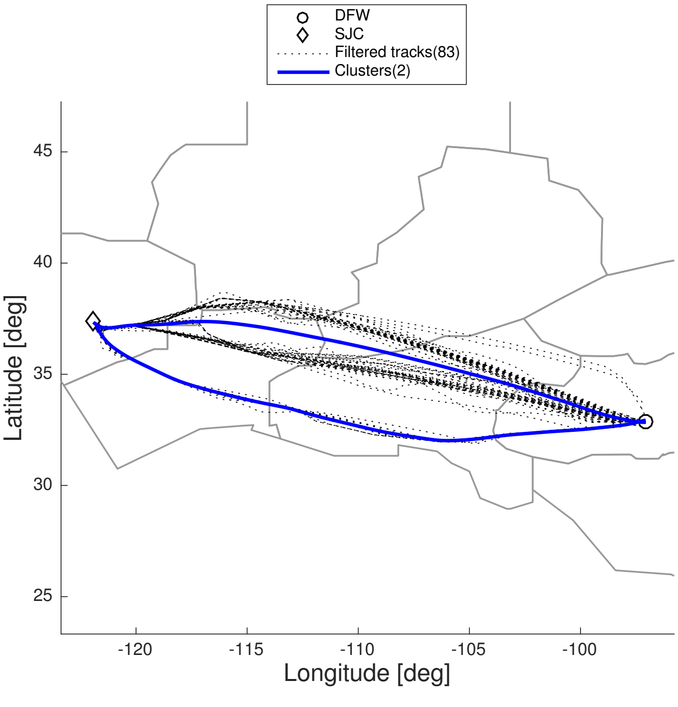

Figure 10.23: SJC, 2 clusters, $\bar{S}_N = 0.597$

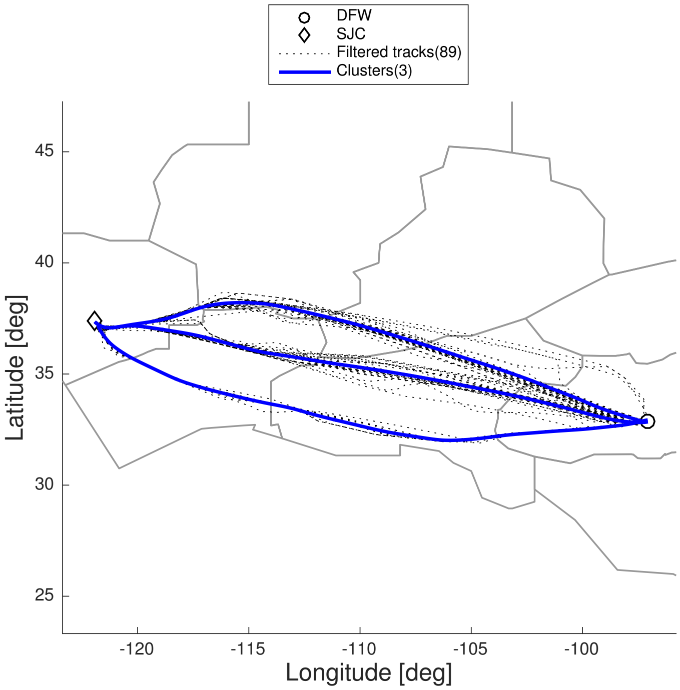

Figure 10.24: SJC, 3 clusters, $\bar{S}_N = 0.668$

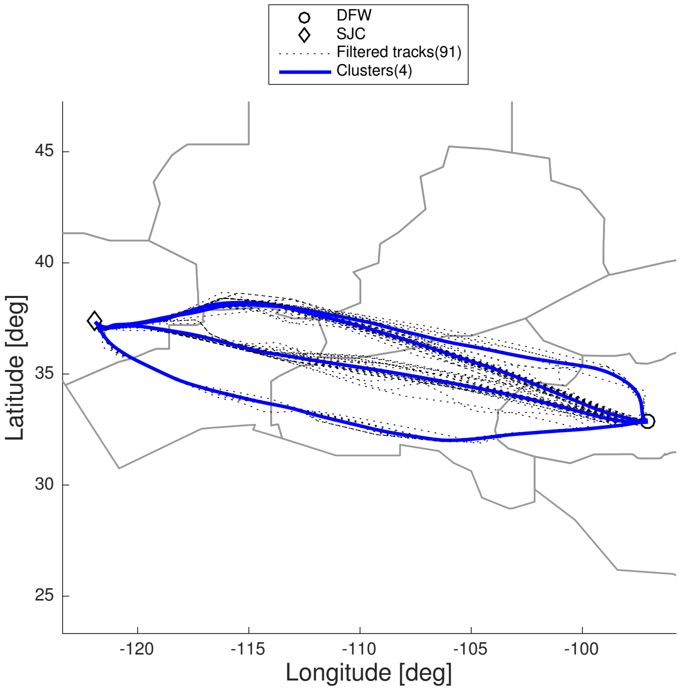

Figure 10.25: SJC, 4 clusters, $\bar{S}_N = 0.636$

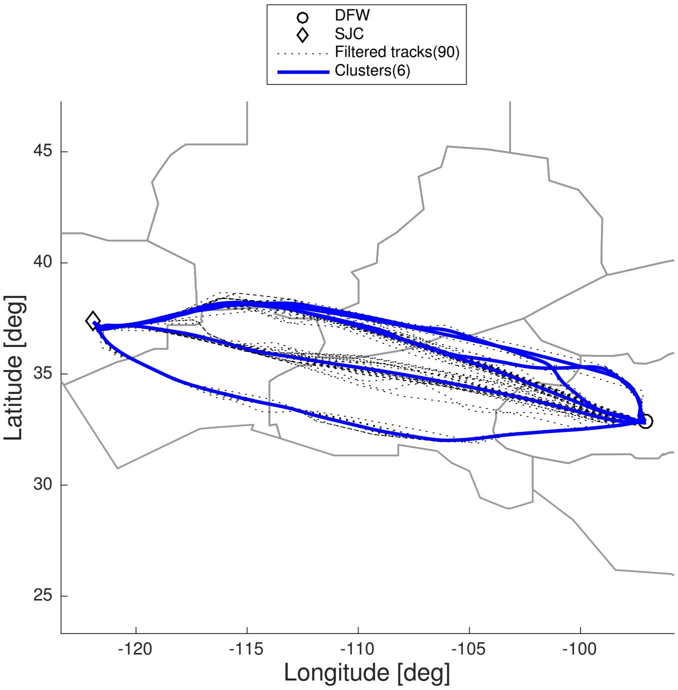

Figure 10.26: SJC, 6 clusters, $\bar{S}_N = 0.568$

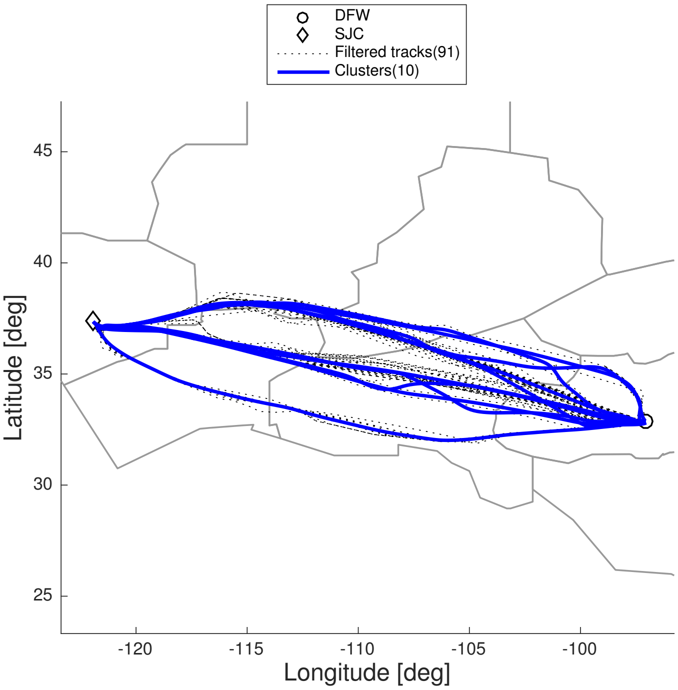

Figure 10.27: SJC, 10 clusters, $\bar{S}_N = 0.291$

Notice that for the 3 cluster distribution, Figure 10.24, the highest $\bar{S}_N$ value is achieved, $\bar{S}_N = 0.668$. This coincides with the expected result by visual analysis. One can calculate $\bar{S}_N$ for all possible number of clusters N_{clust} (from N_r to 2) and obtain the evolution of $\bar{S}_N(N_{clust})$. For this specific example, the result corresponds to Figure 10.28.

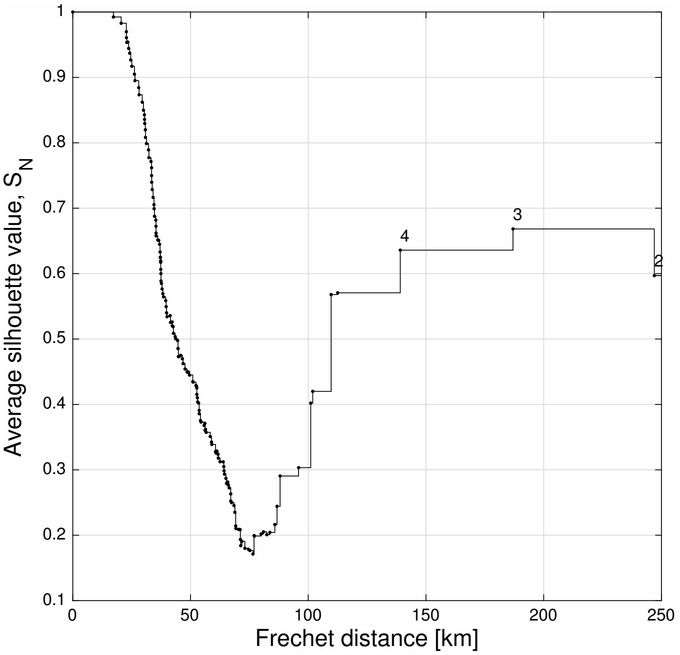

Figure 10.28: Average silhouette score evolution for SJC.

The number of clusters corresponding to each step in the plot is indicated with a number above that step (only for $N_{clust} \leq 4$). The shape of this plot is similar for all scenarios. The first slope, starting at $\bar{S}_N = 1$, reaches a minimum, then the value rises again, reaching a local maximum or stabilizing in the end (for $N_{clust} = 2$). The high values for N_{clust} close to N_r are not meaningful, as they correspond to a distribution where most clusters consist of a single route. It is after the minimum value for $\bar{S}_N$ is passed that representative distributions can be achieved, and the local maximum will represent the most natural or adequate distribution.

As advised in [8] one should not blindly use the average silhouette score as a measure to decide the number of clusters. The presence of severe outliers could result in distributions with single-route clusters corresponding to the outliers having higher average silhouette scores. Thus the outlier filtering step becomes even more important.

In the rest of plots in this work, for convenience, the first part of the evolution of $\bar{S}_N$ is not calculated nor plotted, and a dashed line will replace it. Only the values for $N_{clust} \leq 10$ are calculated.

The whole process (except for the final speed filtering) from the original dataset to the N_{clust} choice has been presented so far. A full example is included next to visualize the process.

LaGuardia Airport (LGA)

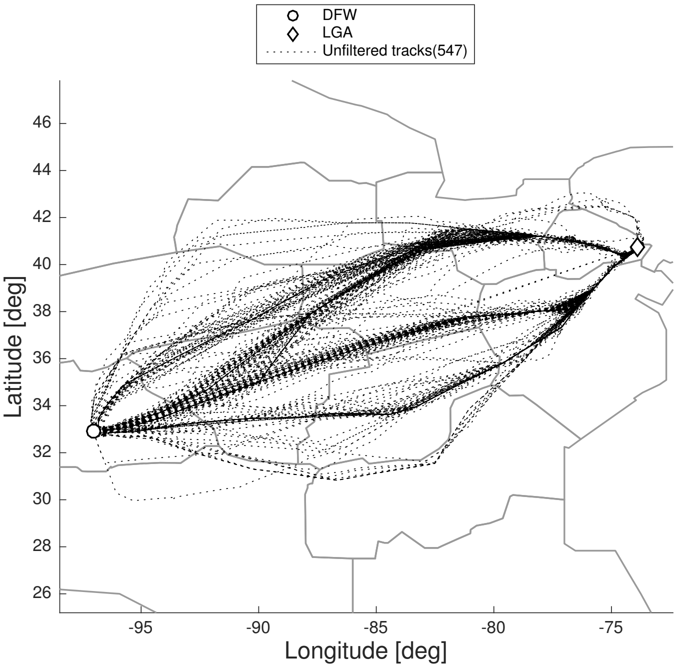

Figure 10.29: Unfiltered tracks, LGA.

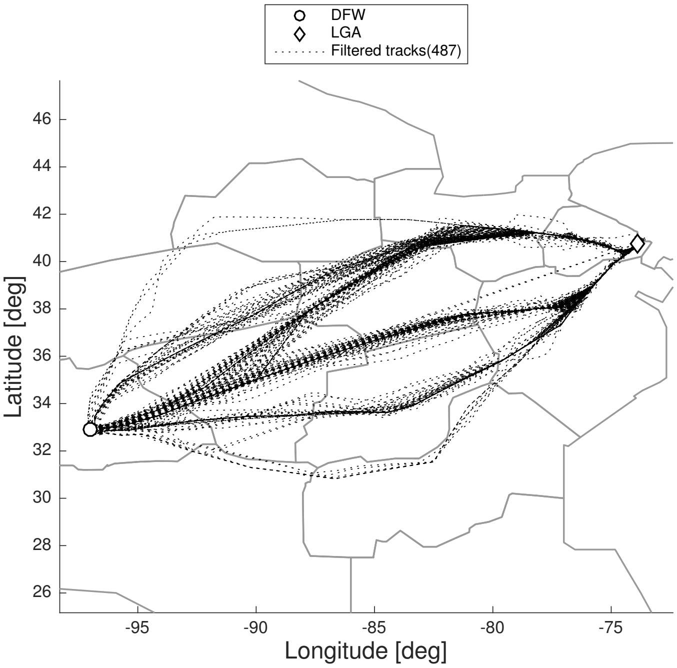

Figure 10.30: Filtered tracks, LGA.

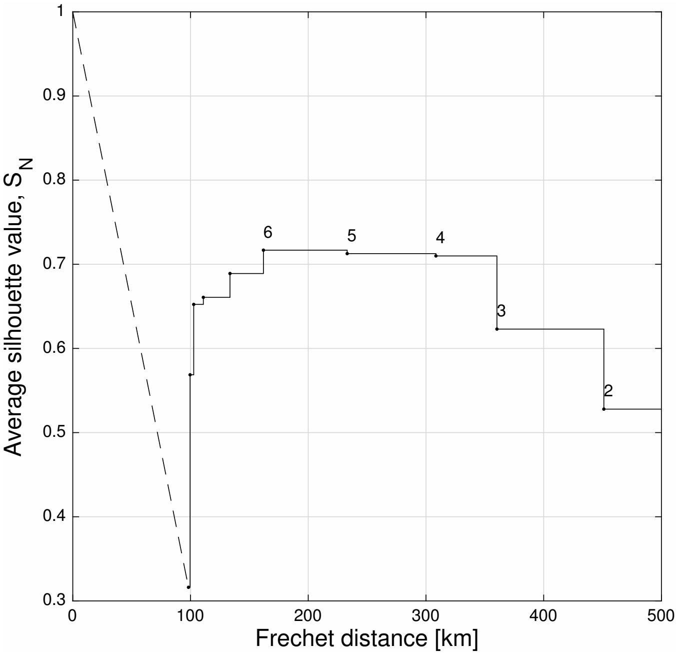

Figure 10.31: $\bar{S}_N$ evolution, LGA.

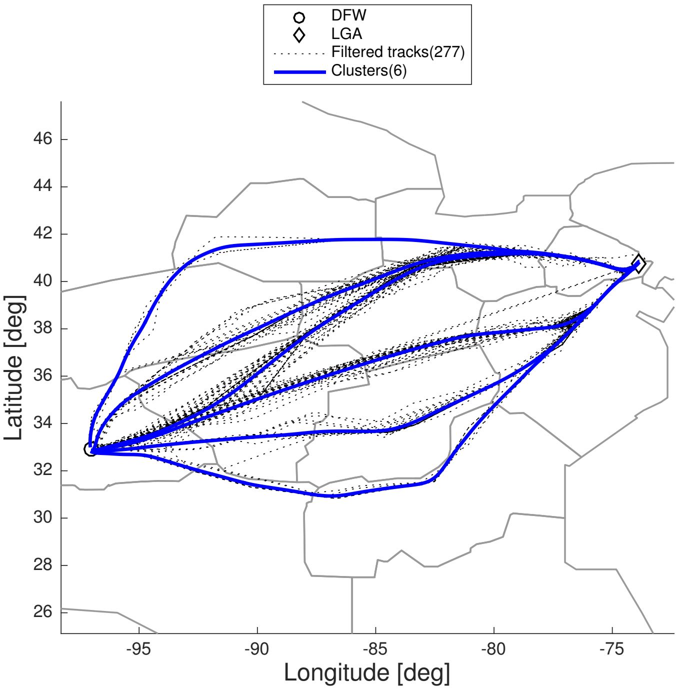

Figure 10.32: Tracks and clusters, LGA.

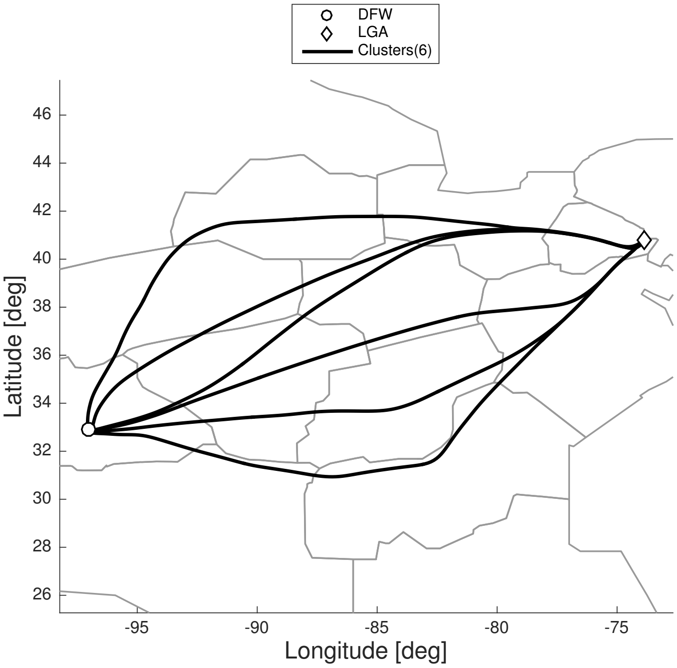

Figure 10.33: Clusters, LGA

The correct behavior of the model is confirmed in this example with complex geometry, internal sub-clusters, cluster density variations and many outliers. The outlier detection algorithm produces good results, and so does the N_{clust} choice method.

One can also see the significance of the obtained flows. For example, the 3 southernmost clusters are operationally very distinct: one is avoiding ZTL center by the norht, the other by the south, while the other is crossing it, corresponding to 3 different operational decisions. The full example of LGA (including the speed filtering and Dip test – presented next – can be found in Appendix A).

Dip Test

The average Silhouette value $\bar{S}_N$ has been introduced as an appropriate quality measure to determine the natural cluster structure in a dataset. However, two or more clusters are required in order to calculate the Silhouette values, as the existence of a 'nearest neighboring cluster' is necessary.

In some situations the routes may form one single cluster, and that solution cannot be determined by maximizing $\bar{S}_N$ as $S(i)$ cannot be calculated. In order to detect single cluster scenarios, the Dip test, introduced by Hartigan and Hartigan in 1984 [28], will be used. The Dip test will distinguish any unimodal from any multimodal distribution.

The *dip statistic* is defined as the maximum difference between the empirical distribution function and the unimodal distribution function that minimizes that maximum difference. A distribution function F is unimodal with mode m if F is convex in $(-\infty, m]$ and concave in $[m, \infty)$. The mode m is not necessarily unique. A unimodal F may have an atom only at a unique mode m, and has a density, except possibly at a unique mode m, that increases in $(-\infty, m)$ and decreases in (m, ∞).

Define

$$\rho(F,G) = sup_x|F(x) - G(x)| \tag{10.8}$$

for any bounded functions F, G. Define

$$\rho(F,\mathcal{A}) = inf_{G\in\mathcal{A}}\rho(F,G) \tag{10.9}$$

for any class $\mathcal{A}$ of bounded functions. Let $\mathcal{U}$ be the class of unimodal distribution functions.

The *dip* of a distribution function F is defined by

$$D(F) = \rho(F,\mathcal{U}) \tag{10.10}$$

Note that

$$D(F_1) \leq D(F_2) + \rho(F_1, F_2) \tag{10.11a}$$

$$D(F) = 0 \text{ for } F\in\mathcal{U} \tag{10.11b}$$

$$D(F) > 0 \text{ for } F\notin\mathcal{U} \tag{10.11c}$$

thus the dip measures departure from unimodality. The maximum value of $D(F)$ is 1/4, achieved when F has two atoms of size 1/2.

In order to determine the probability of a distribution of being unimodal, a significance test is carried out. The dip of the empirical distribution is compared to that of the null distribution through bootstrapping. The appropriate null distribution is uniform (for a deeper discussion and exceptions, see [28]), as the dip is asymptotically larger for the uniform than for any distribution in a wide class of unimodal distributions, those with exponentially decreasing tails. For a given empirical distribution, a random uniform distribution with the same sample size as the empirical distribution is generated, and its dip is calculated. This process is repeated a set number of times *nboot* (500 for all shown examples) and all dip values for the uniform distributions are stored in vector $boot_{dip}$. With dip= dip of the empirical distribution,

$$p = \frac{count(dip < boot_{dip})}{nboot} \tag{10.12}$$

so p corresponds to the ratio of cases when the dip of the uniform distribution is greater than that of the empirical distribution, versus the total number of tested uniform distributions. Equivalently, it represents the probability of the empirical distribution of being unimodal.

The following figure illustrates the behavior and results of the dip test for two randomly generated unimodal distributions of the same sample size, with modes that are initially coincident but which are slowly separated so the distribution becomes progressively bimodal.

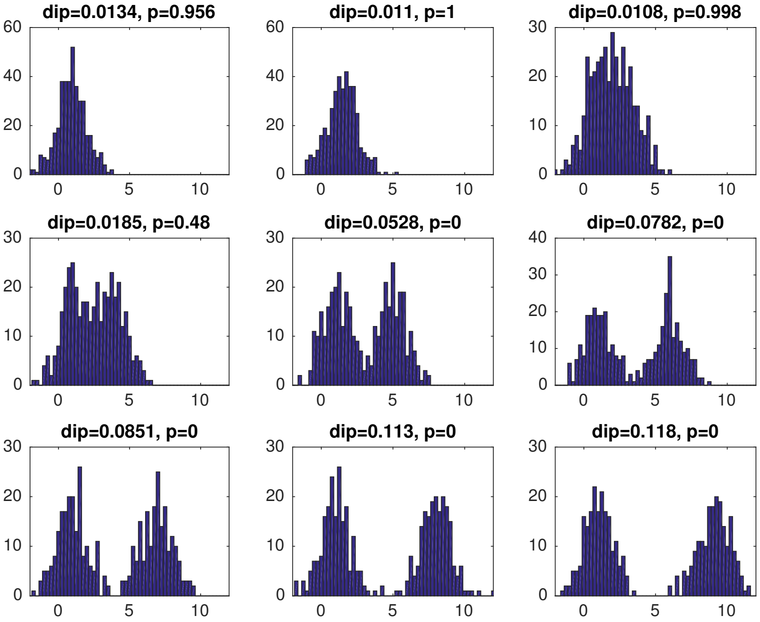

Figure 10.34: Behavior of dip test for different distributions.

In order to use the dip test, for each distribution of routes, one of the two extreme routes (meaning one of the two routes situated at the boundaries of the route distribution) is found, and the Fréchet distances of all other routes with respect to this route are used as data points for the empirical distribution.

Because the dip test is very sensitive (see Figure 10.34), a threshold value of $p = 0.75$ will be used to determine whether or not a distribution is unimodal. A distribution will be considered unimodal for values of $p > 0.75$. The general process followed to determine the number of clusters of a route distribution is the following.

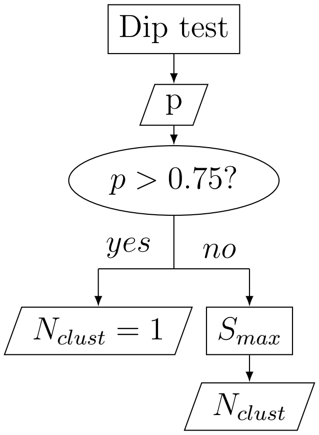

Figure 10.35: N_{clust} selection process.

Following are some examples of unimodal route distributions and the results of the performed dip test.

McCarran International Airport (LAS)

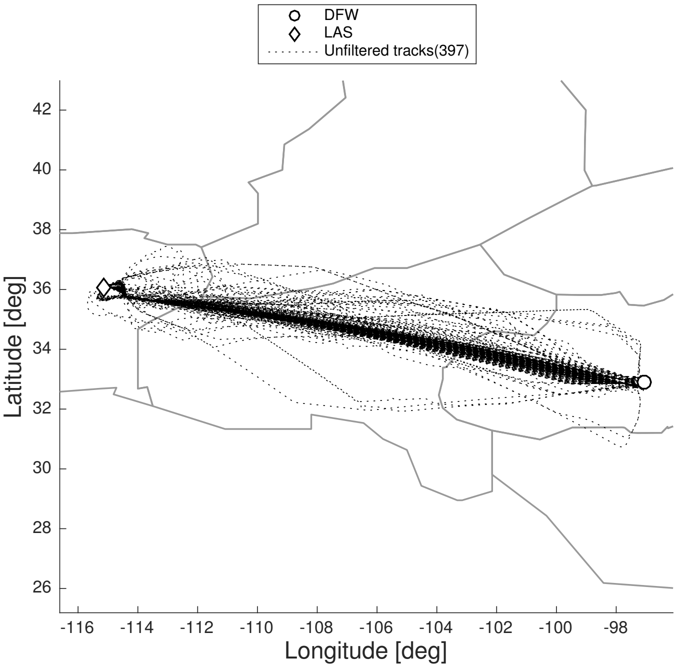

Figure 10.36: Unfiltered tracks, LAS.

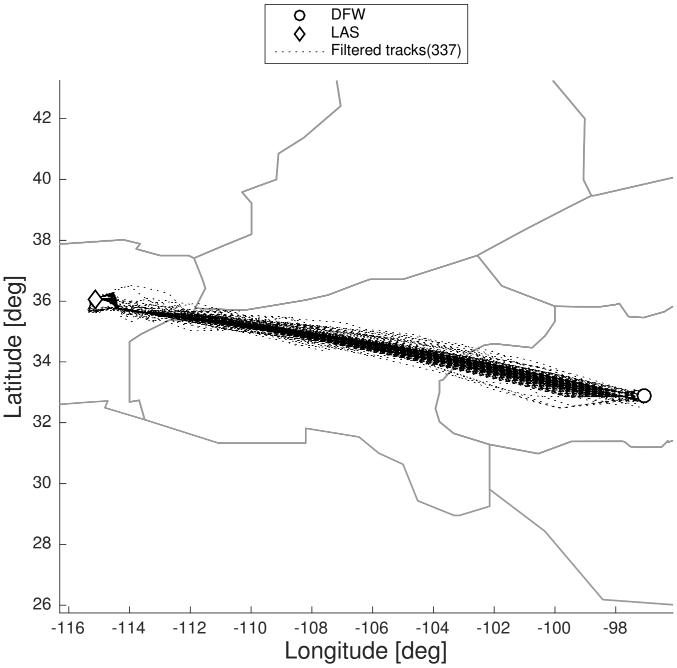

Figure 10.37: Filtered tracks, LAS.

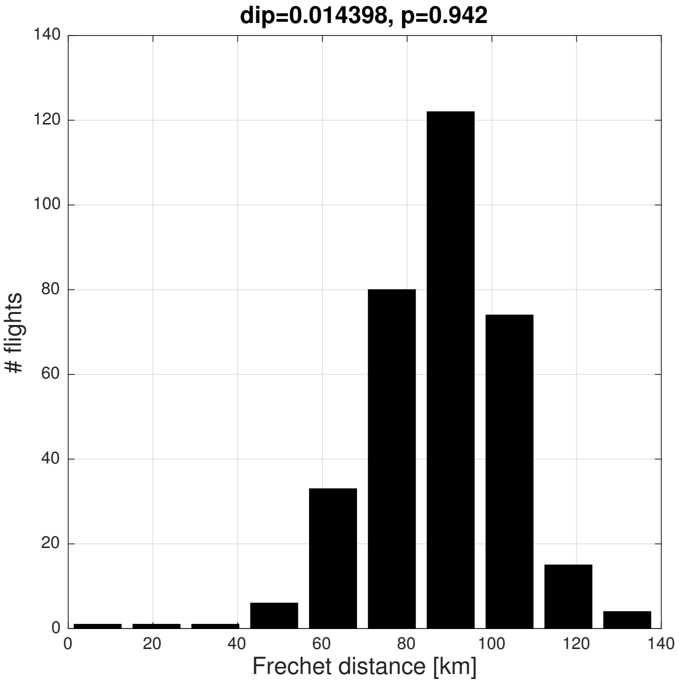

Figure 10.38: Dip test, LAS.

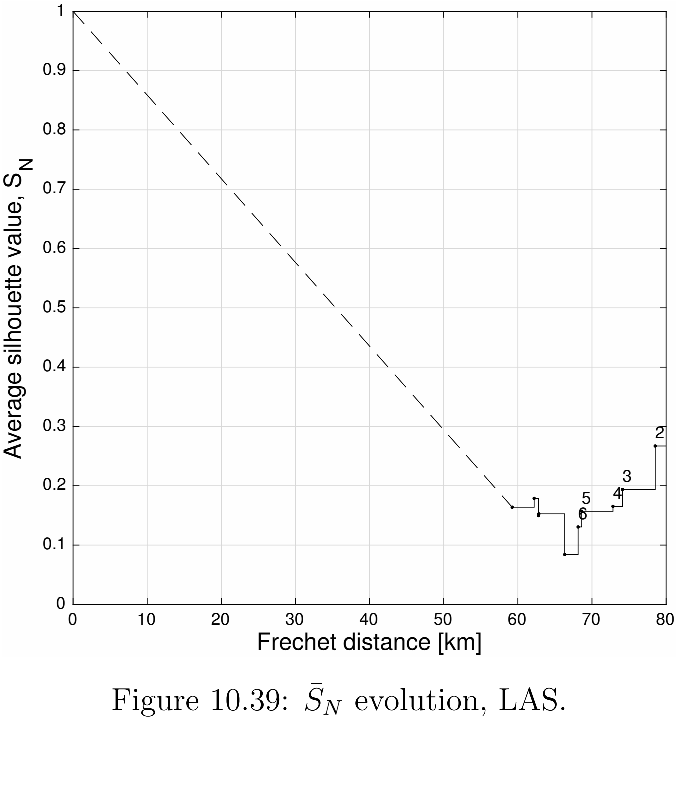

Figure 10.39: $\bar{S}_N$ evolution, LAS.

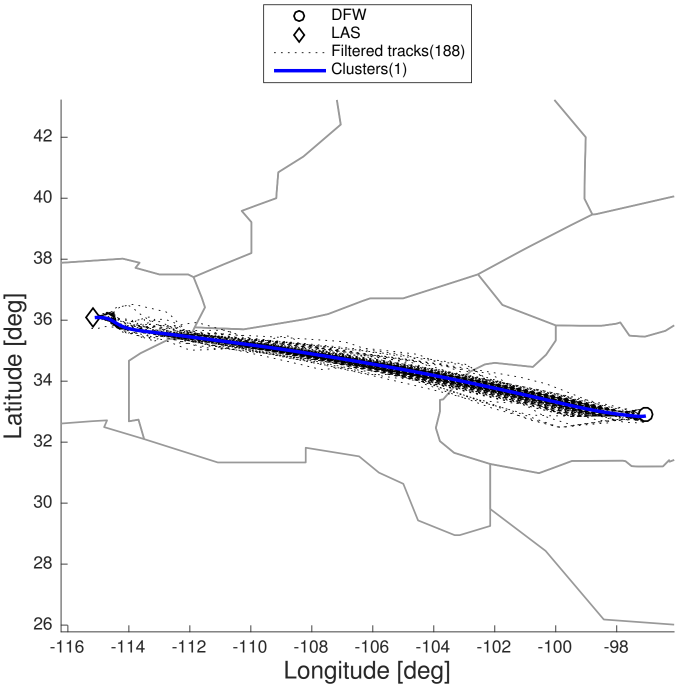

Figure 10.40: Tracks and clusters, LAS.

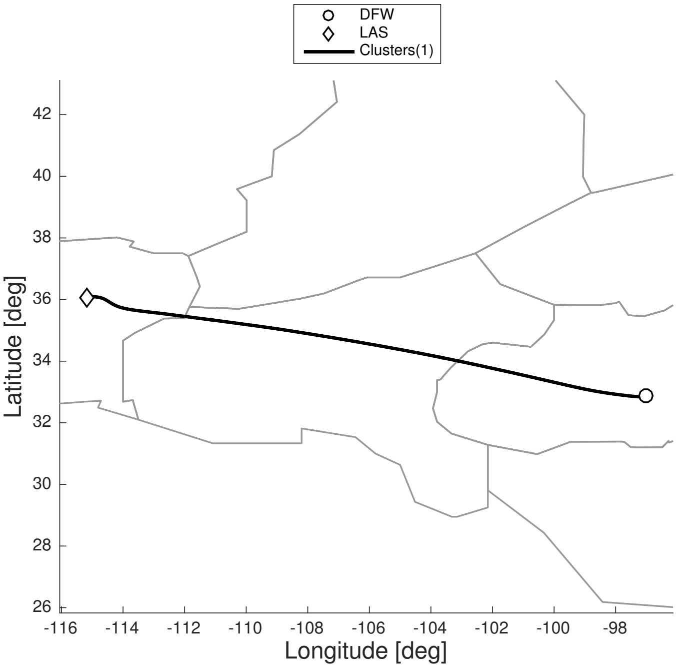

Figure 10.41: Clusters, LAS

Eppley Airfield (OMA)

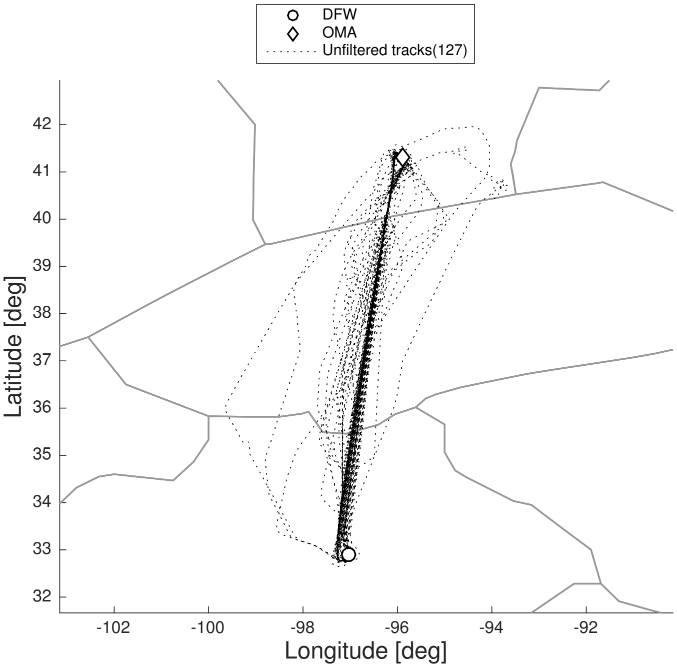

Figure 10.42: Unfiltered tracks, OMA.

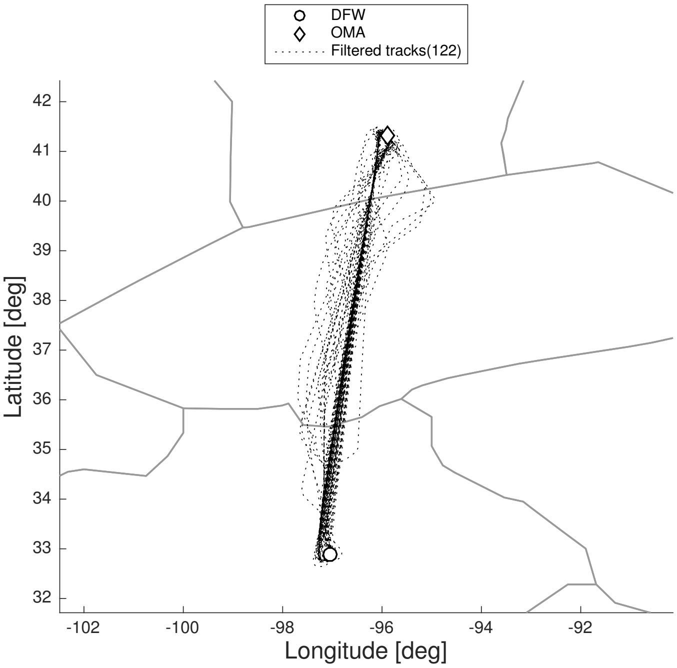

Figure 10.43: Filtered tracks, OMA.

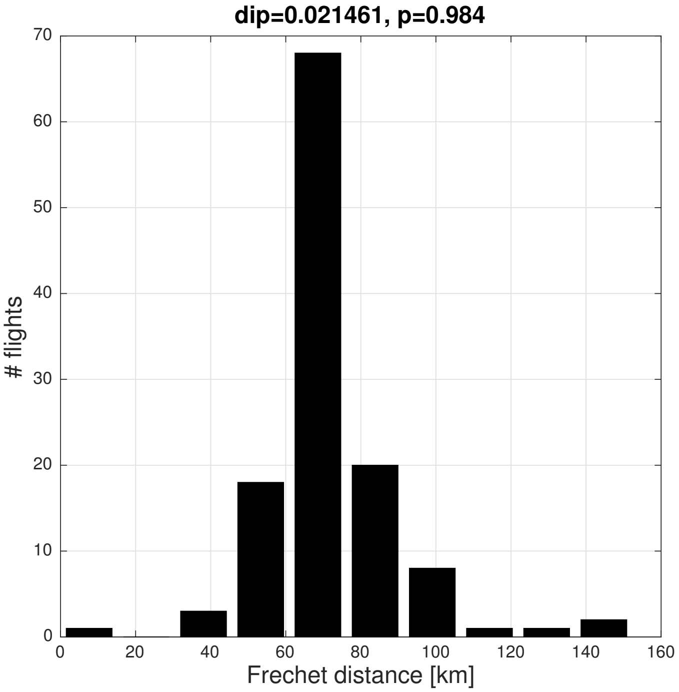

Figure 10.44: Dip test, OMA.

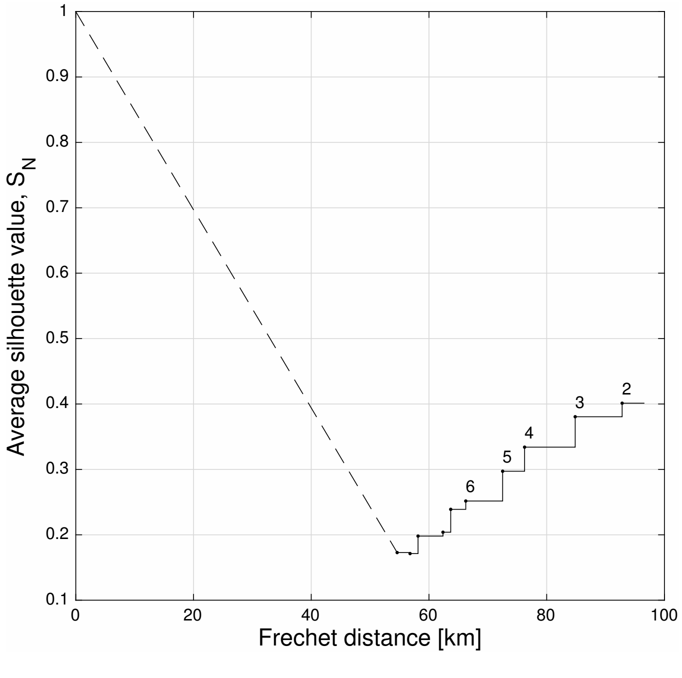

Figure 10.45: $\bar{S}_N$ evolution, OMA.

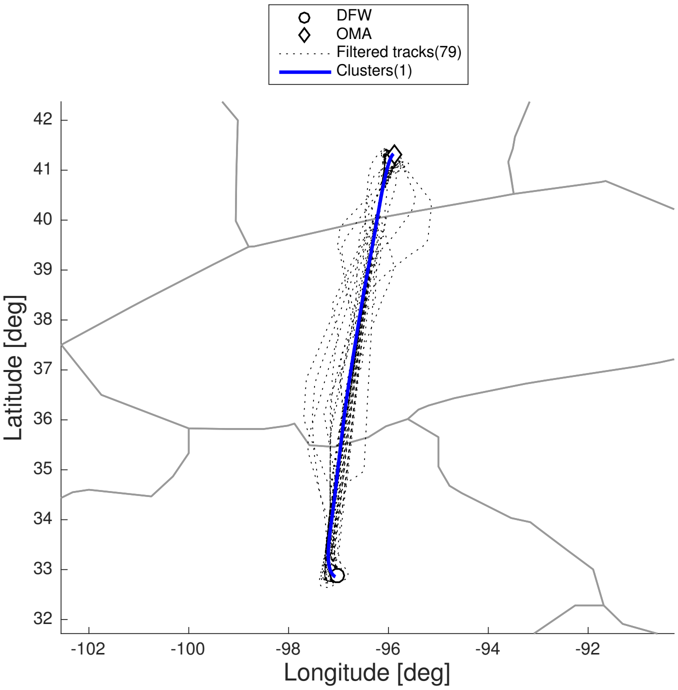

Figure 10.46: Tracks and clusters, OMA.

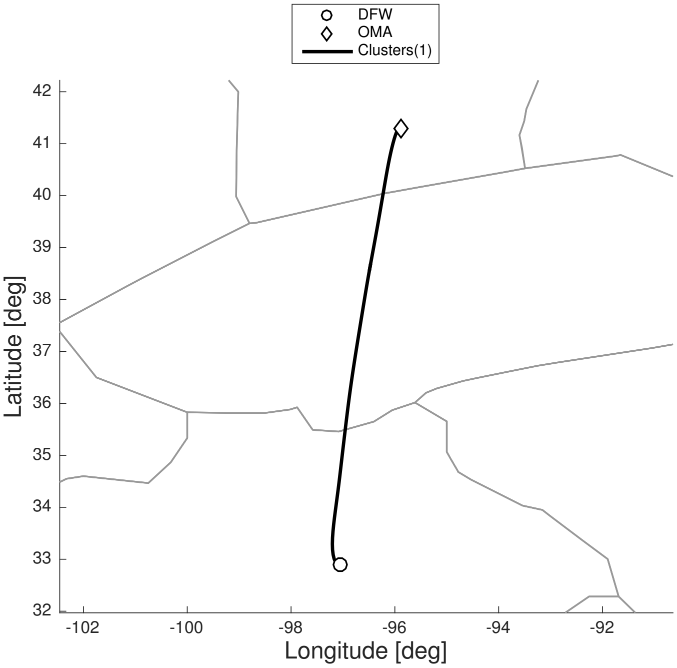

Figure 10.47: Clusters, OMA

The histogram of the empirical distributions is represented in Figure 10.38 and Figure 10.44, together with the *dip* value and the p value. In both cases, we see p values of over 0.9, which correctly characterizes both route distributions as unimodal. In addition, Figure 10.39 and Figure 10.45 show the low values of $\bar{S}_N$ for the rest of N_{clust} alternatives. Note that having low $\bar{S}_N$ values, on its own, does not imply that the distribution must be unimodal, as some route distributions may have very disperse clusters in which case all $\bar{S}_N$ values will be relatively low.

10.7 Speed Filtering and Aggregate Route Creation

The reader may have noticed that after the outlier filtering step, and when the aggregate routes are added, the number of filtered routes decreases. This is due to the last filtering before generating the aggregate routes: the speed filtering.

The need for this step will be understood after explaining the process by which the aggregate routes are created; once the number of clusters for a route distribution has been chosen using the Dip test and the S_{max} calculations, it's time to characterize each cluster by an *aggregate route*, or a single route that best represents that cluster of routes.

Because this network is being generated in order to be used in a dynamic model, both the geometry of the aggregate route as well as the speed at which it is traveled must be representative of the routes in that cluster. Each route is characterized by a sequence of points that are taken at constant time steps (i.e. the time required to travel from any point in a route to the next point is always the same). Therefore, the distribution of the points in a route is representative of the instantaneous speed of the aircraft flying it.

The approach to generate the aggregate network and be geometrically and dynamically representative of the cluster of routes is to determine the nodes of the aggregate route as the centroids of the corresponding nodes of all routes in that cluster. In this way, the first node of the aggregate route will correspond to the centroid of all the first nodes of all routes in that cluster, and subsequently for the rest of nodes.

An alternative approach was tested in which the aggregate route for each cluster was chosen as the single route in that cluster that minimized the average Fréchet distance from it to the rest of routes in the cluster. This approach was found unsuitable because the aggregate route was strongly dependent on the features of one single route. If the chosen route had an unrepresentative geometry or speed profile, or it had local features that didn't represent the cluster, the aggregate route obtained was not an appropriate choice. Thus, the approach previously described is more robust as all features are averaged.

A problem arises when routes in a cluster have different number of nodes (due to speed or geometrical differences), which is granted to happen. At some point during the creation of the aggregate route, some routes will have reached the destination airport while some other routes will not, and the subsequent nodes of the aggregate network from this point on will be incorrectly placed. This problem is repeated every time a route reaches the destination and the clustering continues, resulting in inconsistent aggregate routes. Following is an example of the aggregate routes arriving at Pittsburgh, created without speed filtering, to visualize the described effect.

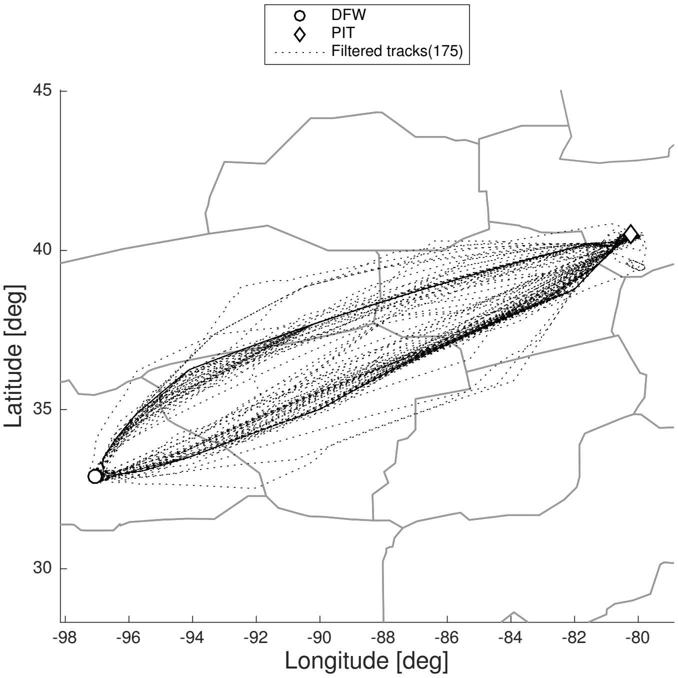

Figure 10.48: Tracks without outliers, PIT.

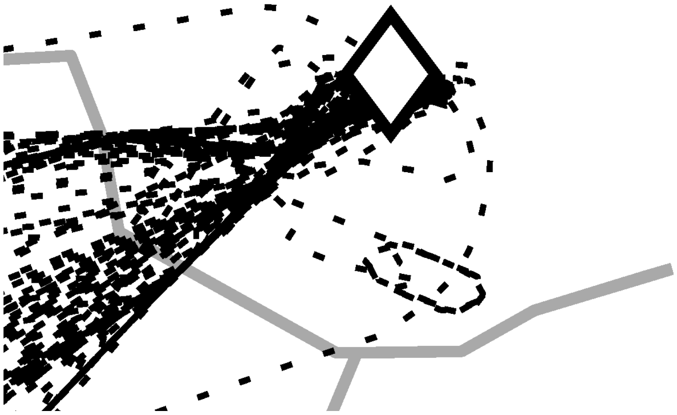

Figure 10.49: Arrival to PIT. No speed filtering.

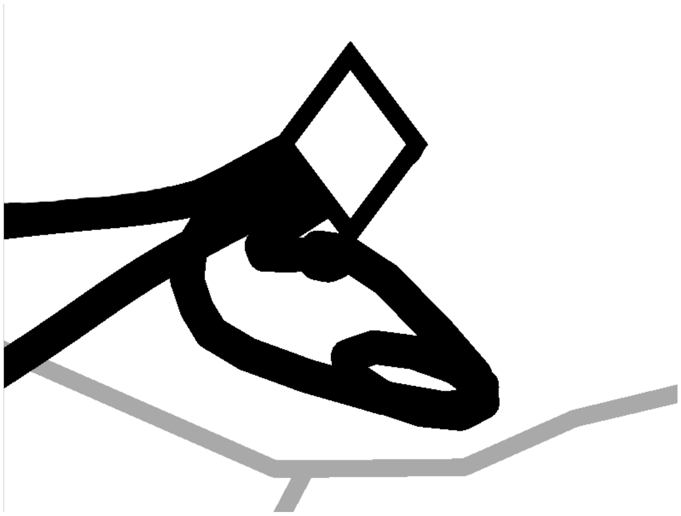

Figure 10.50: Aggregate routes arriving at PIT. No speed filtering.

A hippodrome wait pattern can be clearly identified in one of the aggregate routes. This is caused by a flight that is very similar to all other flights (so it was not discarded as an outlier) but is longer than the rest because of the waiting pattern before landing. Thus, the extra nodes of this route considerably affect the resulting aggregate route. This effect also occurs when some routes are slightly dissimilar in number of nodes, not necessarily due to a waiting pattern which is an extreme case.

In order to solve this problem, once the clusters have been formed, the number of nodes of all routes in a cluster will be made equal. The first step is to eliminate the routes with too many or too few nodes, because they would not be adapted correctly to the new imposed number of nodes. This is done by sorting the routes based on their number of nodes, and eliminating the first and last quartiles of the distribution. Once this is done we are left with the 50% of the original routes in that cluster, but the quality of the cluster will be improved due to the increase in dynamic consistency. In addition, this step also improves the geometric consistency of the clusters, as the eliminated routes are usually those that are geometrically more dissimilar as well. Then, out of the left routes, the route with the least number of nodes $(n_n)_{min}$ is found. For each of the other routes, nodes are randomly eliminated until only $(n_n)_{min}$ nodes are left. Now, all routes have the same number of nodes. The effects of the random deletion of nodes is compensated between routes. This process is only carried out if the cluster has at least 10 routes.

Notice that this process is conservative in terms of speed of the aggregate route (the speed is underestimated), as the aggregate route speed will correspond to the first quartile of the speed distribution of the routes in that cluster. This limit could be modified.

The effect of the speed filtering can be observed in Figure 10.53.

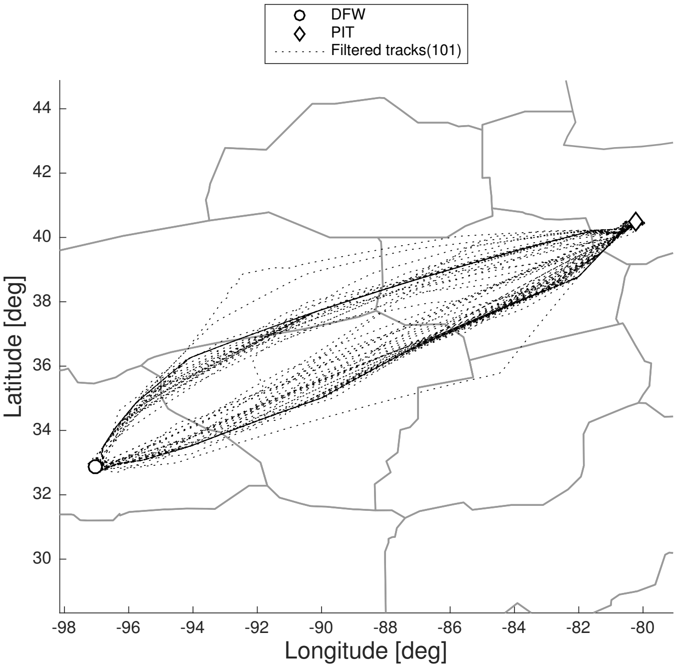

Figure 10.51: Tracks without outliers after speed filtering, PIT.

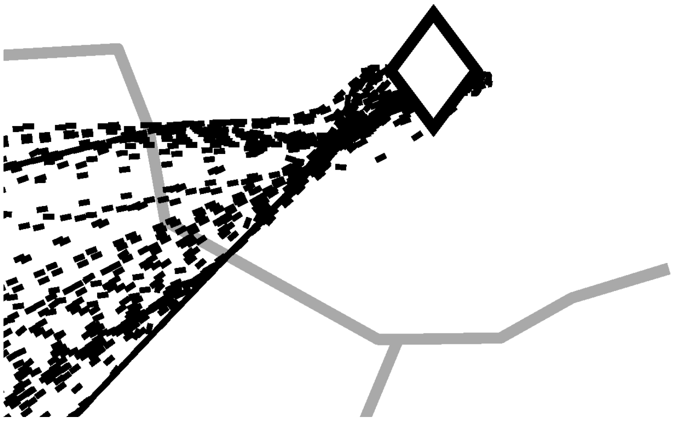

Figure 10.52: Arrival to PIT. Speed filtering.

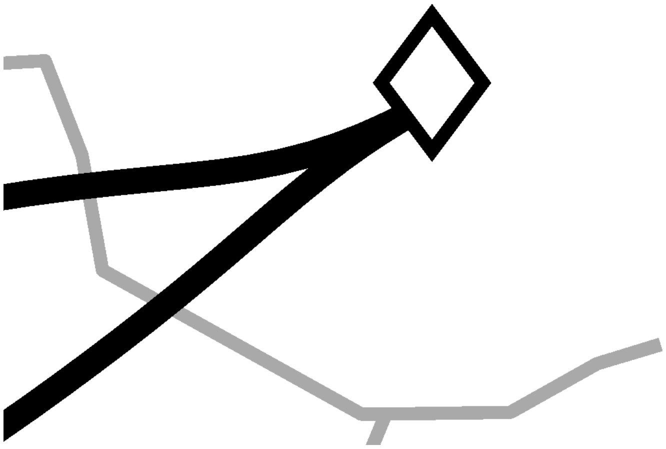

Figure 10.53: Aggregate routes arriving at PIT. Speed filtering.

The histograms for the number of nodes of the various routes in each cluster are included next. The vertical dashed lines delimit the first and third quartile of the distribution, and all routes on the extremes delimited by these lines are eliminated before clustering. The number of nodes indicated by Q_1 is the one used for the corresponding agregate routes. The values of the quartiles are added in each plot. In this context, n_n is the number of nodes of a route.

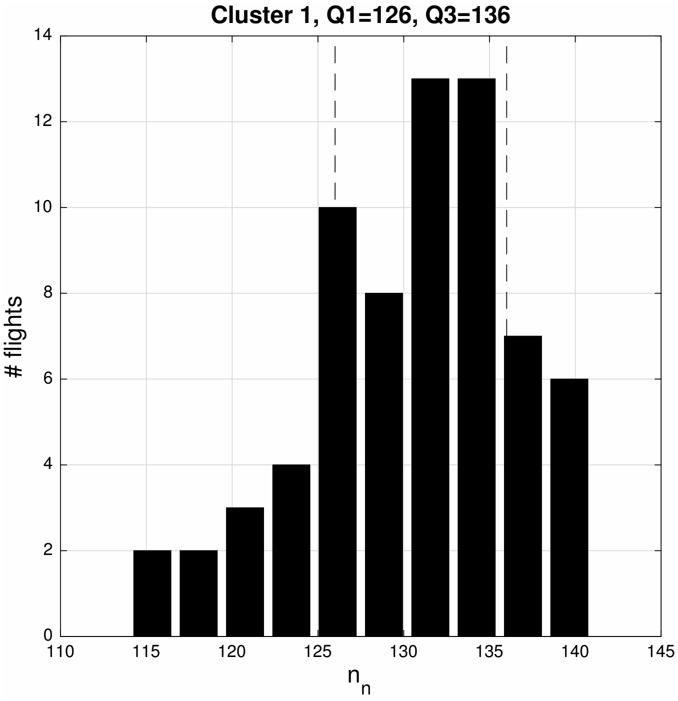

Figure 10.54: Histogram for speed filtering. Cluster 1, PIT.

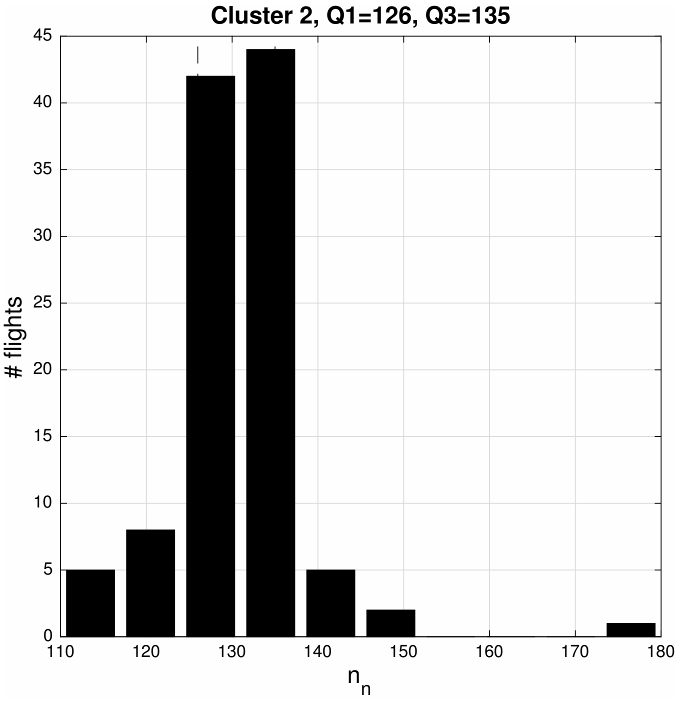

Figure 10.55: Histogram for speed filtering. Cluster 2, PIT.

For scenarios with more disperse clusters (like LAX) the histograms of n_n become similar to normal distributions (or, in general, unimodal distributions). See the following examples,

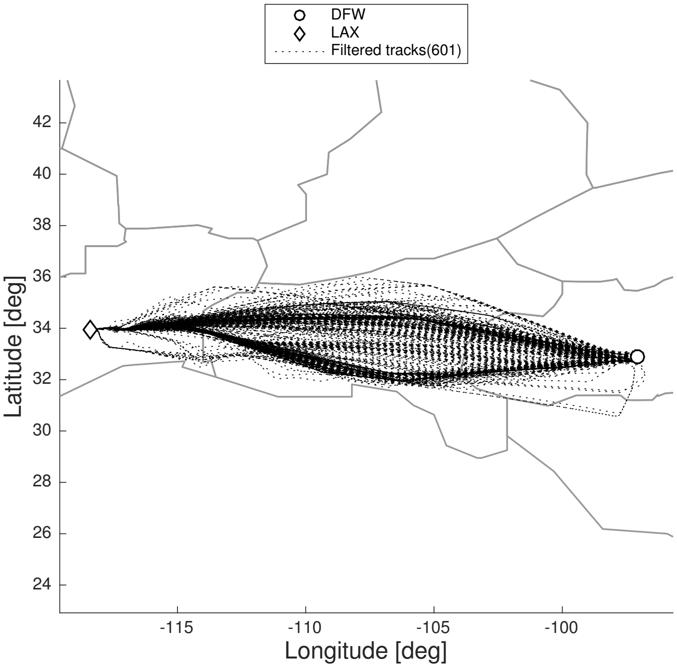

Figure 10.56: Tracks without outliers, LAX

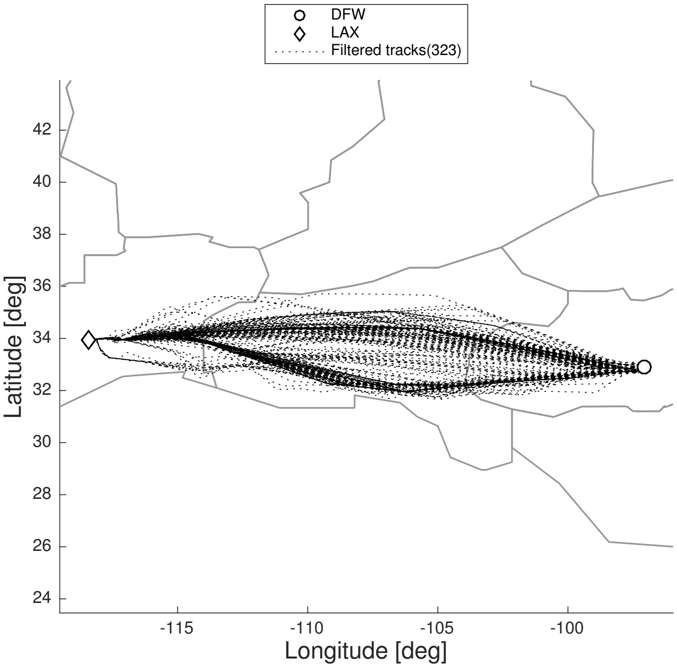

Figure 10.57: Tracks without outliers after speed filtering, LAX.

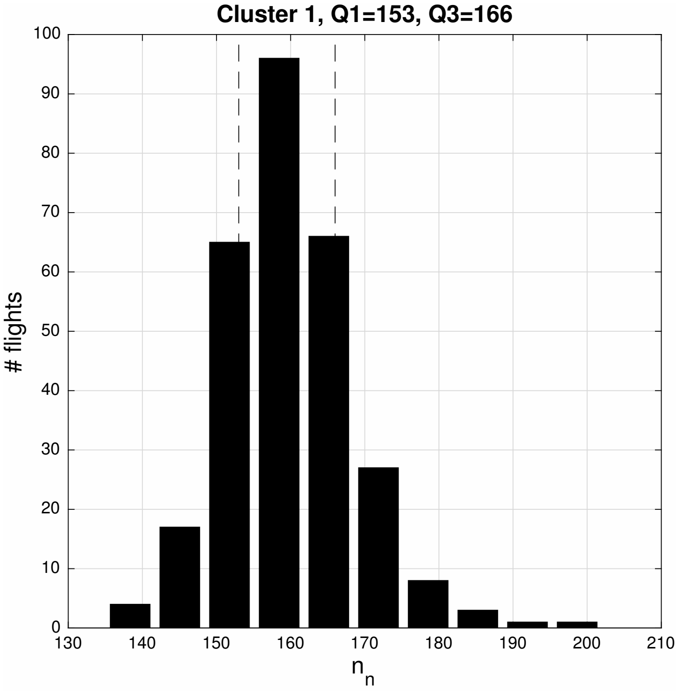

Figure 10.58: Histogram for speed filtering. Cluster 1, LAX.

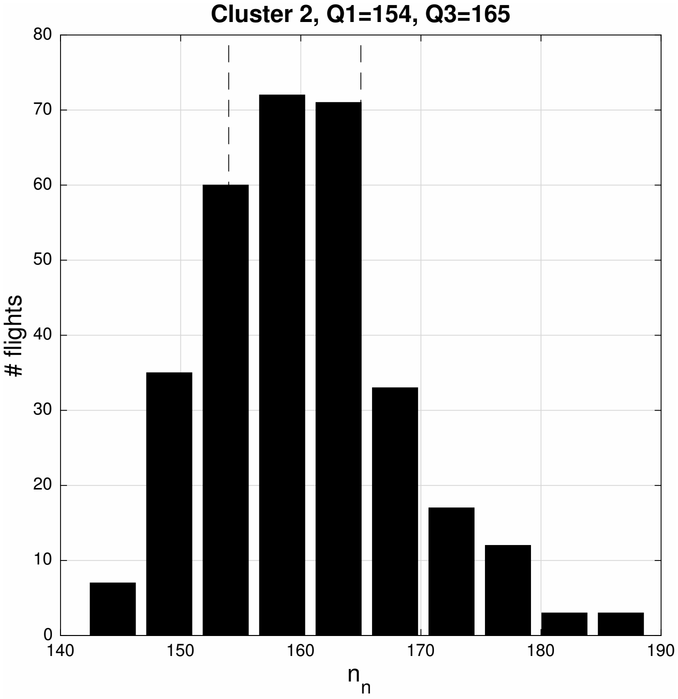

Figure 10.59: Histogram for speed filtering. Cluster 2, LAX.

10.8 Performance of S_{max} Method

It is interesting to first compare the performance of the developed method with that of the fixed threshold and the descriptive function. Lets start by looking at the behavior of the S_{max} method in the cases presented in *Section 10.5 Problem of the Presented Methods*.

Seattle-Tacoma International Airport (SEA)

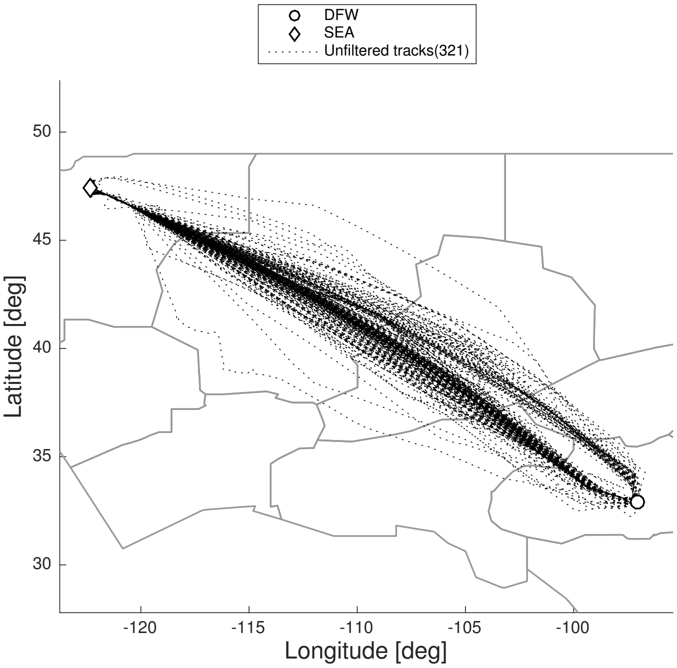

Figure 10.60: Unfiltered tracks, SEA.

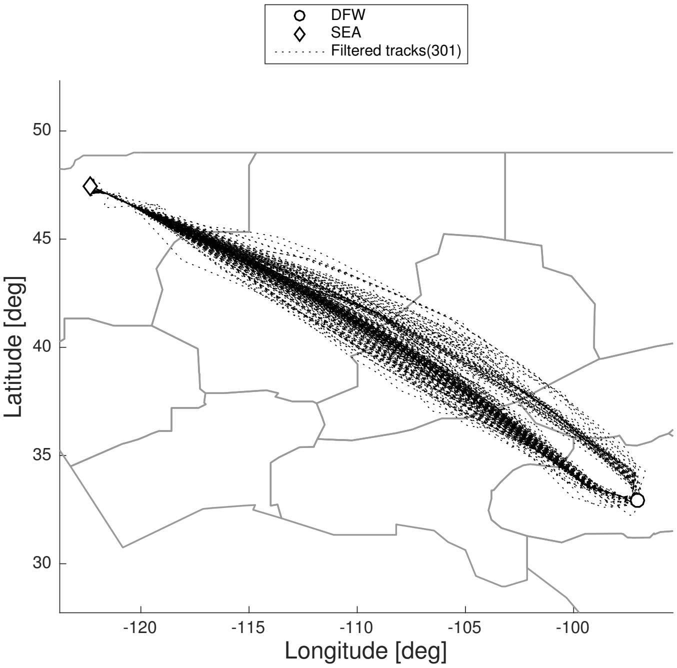

Figure 10.61: Tracks without outliers, SEA.

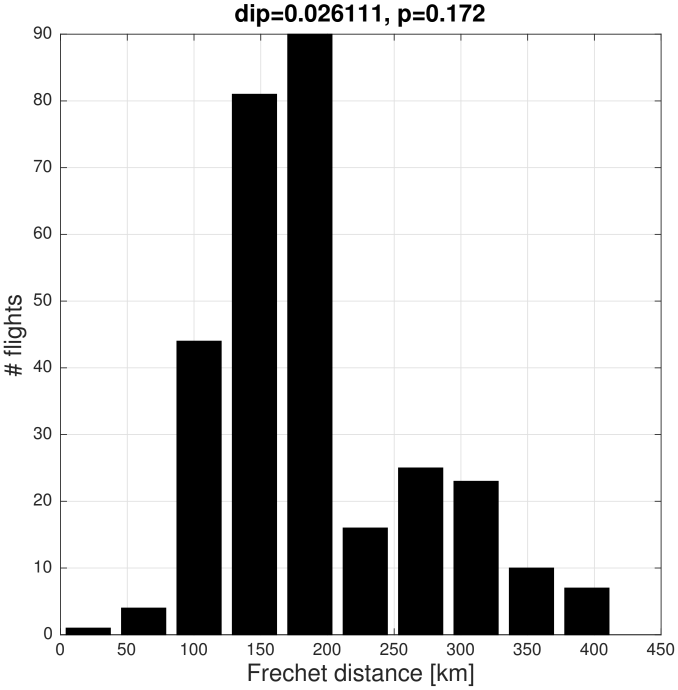

Figure 10.62: Dip test, SEA.

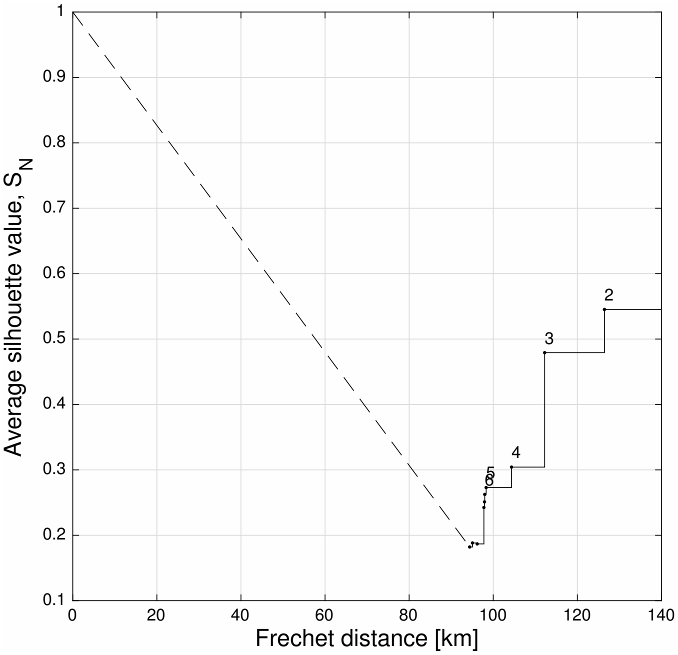

Figure 10.63: $\bar{S}_N$ evolution, SEA.

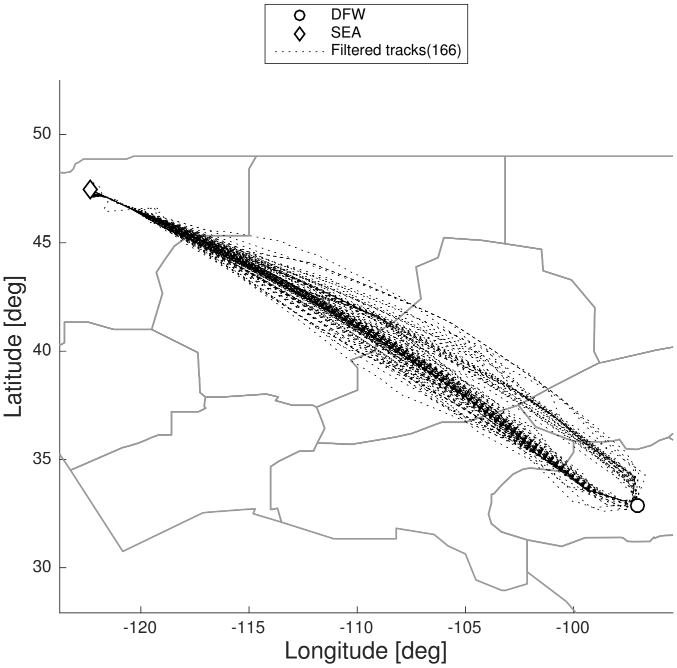

Figure 10.64: Tracks without outliers after speed filtering, SEA.

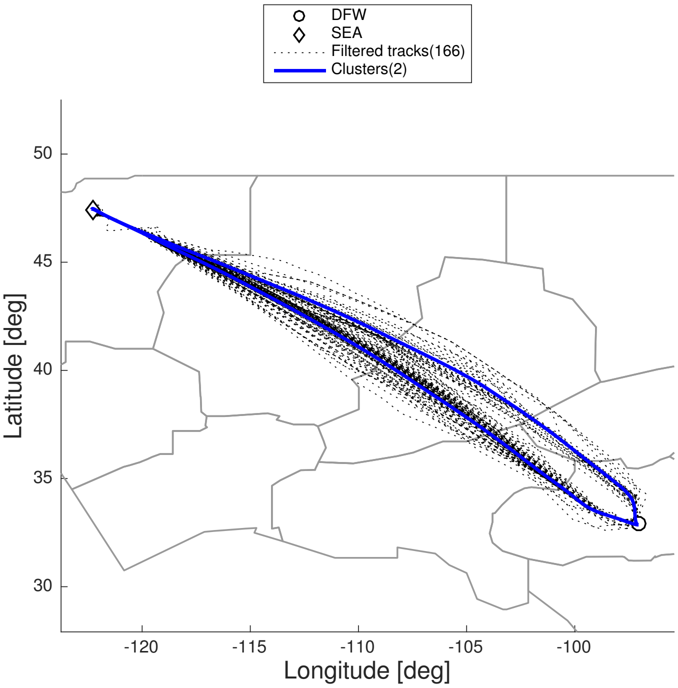

Figure 10.65: Final tracks and clusters, SEA.

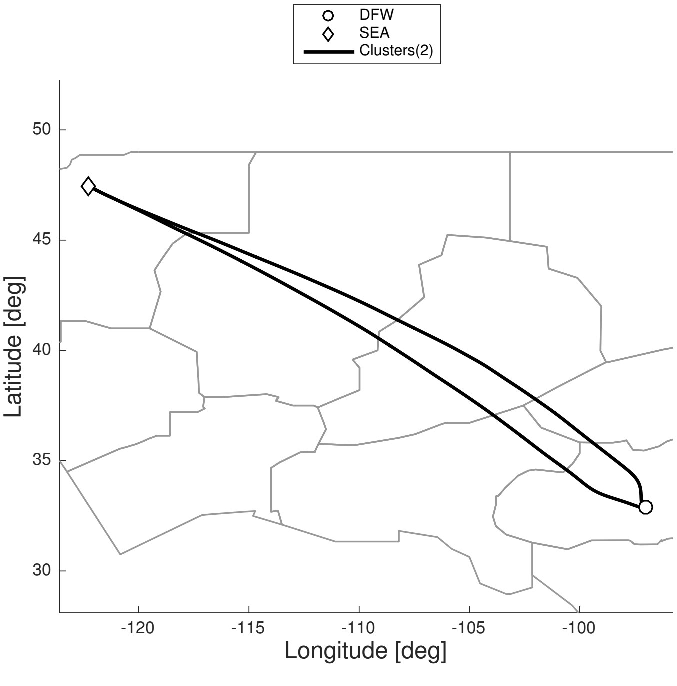

Figure 10.66: Clusters, SEA

Pittsburgh International Airport (PIT)

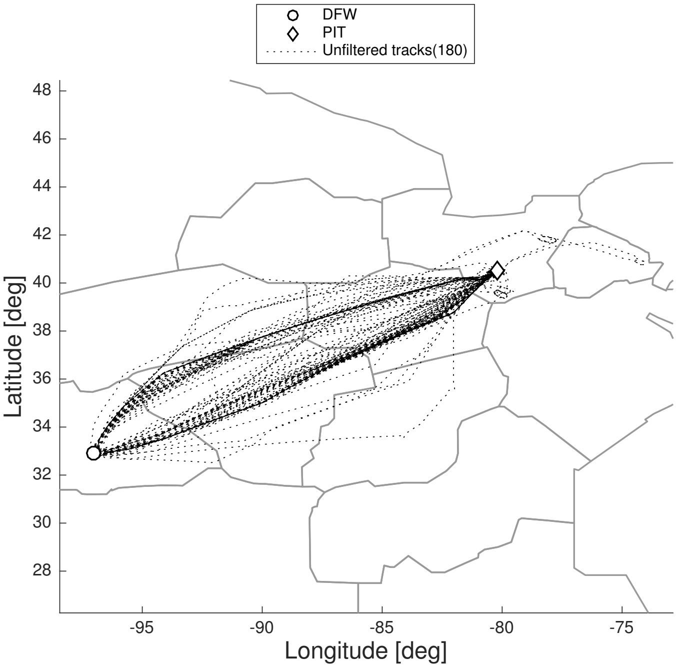

Figure 10.67: Unfiltered tracks, PIT.

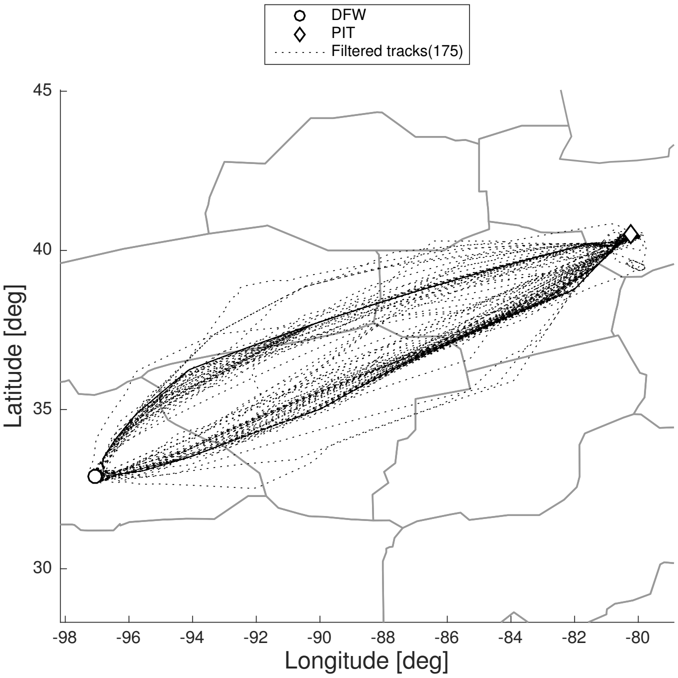

Figure 10.68: Tracks without outliers, PIT.

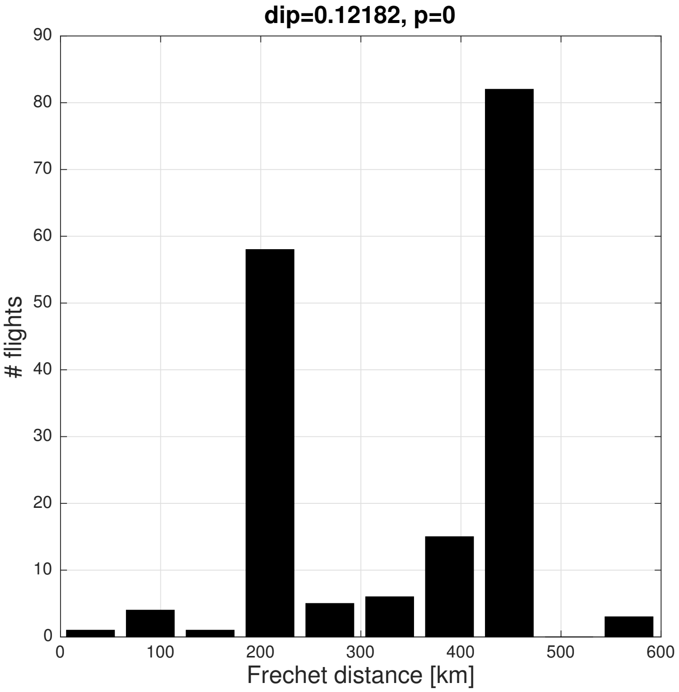

Figure 10.69: Dip test, PIT.

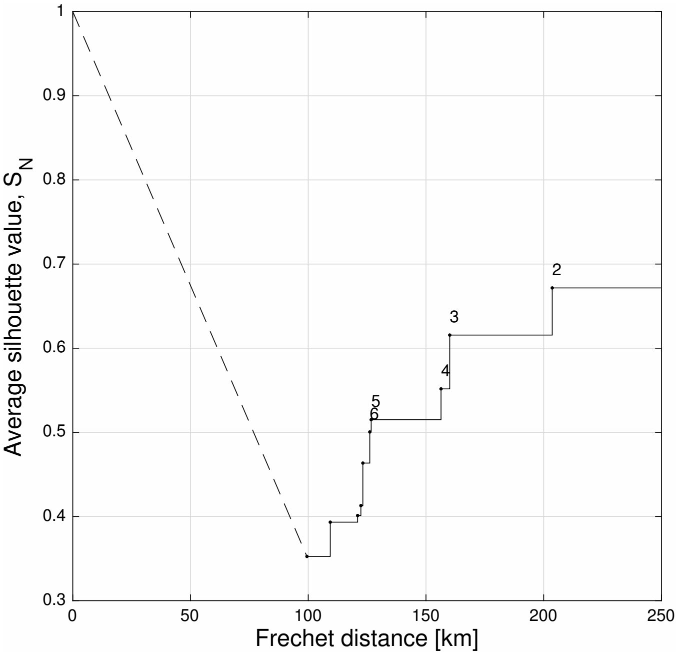

Figure 10.70: $\bar{S}_N$ evolution, PIT.

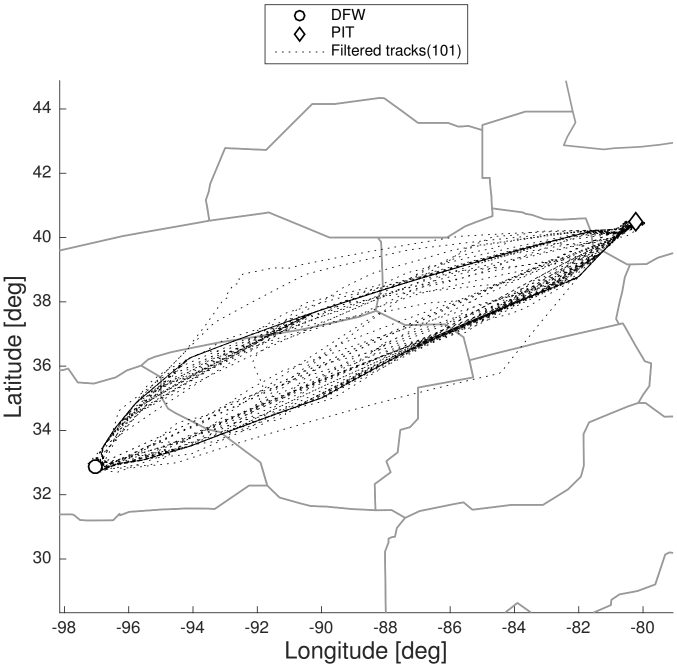

Figure 10.71: Tracks without outliers after speed filtering, PIT.

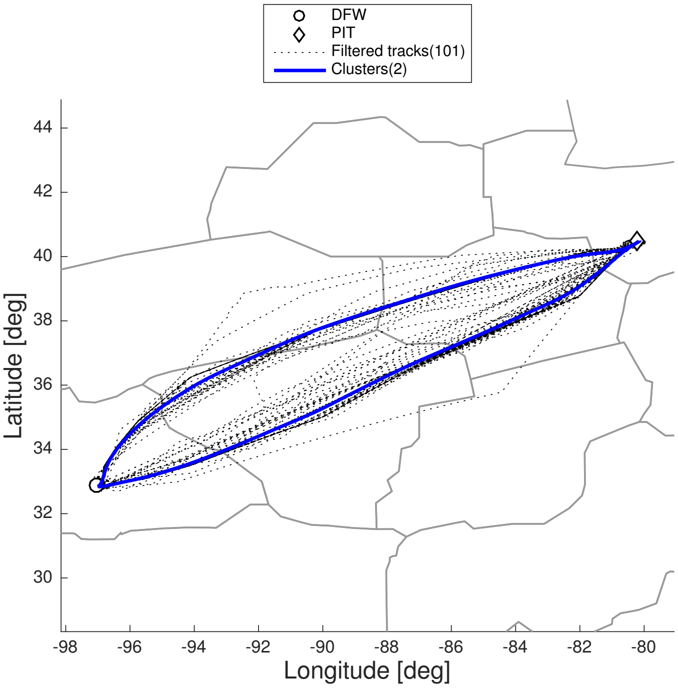

Figure 10.72: Final tracks and clusters, PIT.

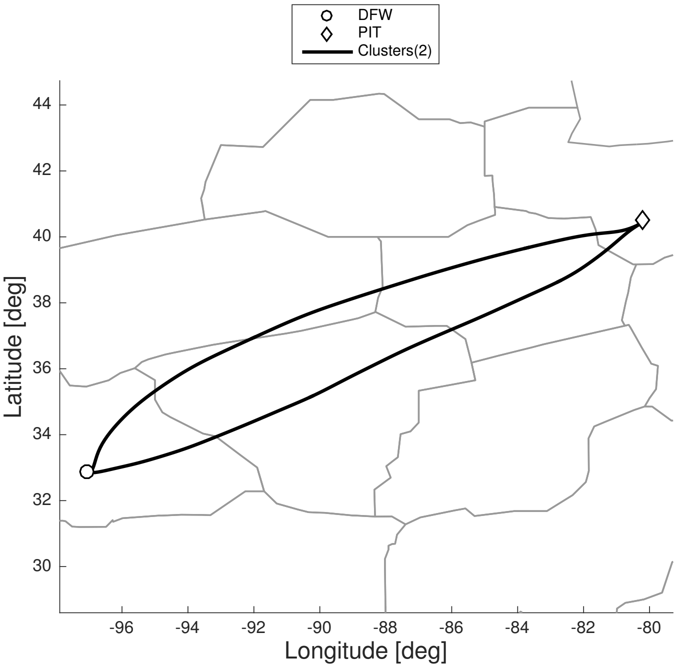

Figure 10.73: Clusters, PIT

The developed method correctly predicts the expected number of clusters in both cases. Remember that these cases are specially challenging due to the differences in the dendrogram structures and the correct pruning heights, and none of the other methods had the ability to correctly predict both results.

The success rates for the developed method have been calculated too, and are presented next together with the success rates of the alternative methods.

	D_t	**K**	S_{max}
Single	51	49	72
Complete	62	68	79
Average	62	66	81

Table 10.3: Success rate [%] of different pruning methods.

The increase in performance using the developed method is considerable. It is important to remark that the success rates are not free from error, as the target values are assigned after visual analysis of the scenarios. However, the target configurations – after being determined – are the same for all tests, so even though these exact success rate values may not be totally accurate, the tendencies shown should be.

The average cophenetic coefficients for every linkage method have also been calculated (they are independent from the pruning method),

	Single	Complete	Average
$\bar{\boldsymbol{C}}$	0.80	0.83	0.89

Table 10.4: Average cophenetic coefficient for different linkage methods.

Chapter 11

Conclusions

An automatic method for route clustering with outlier detection has been developed combining the Silhouette score and the Dip test as cluster quality measures and decision indicators. The method has been compared to two other alternatives, and an increase in flexibility is apparent.

The presented method performs well in a wide range of situations, with heavy cluster density variations, presence of outliers, disperse clusters, internal sub-cluster structures and complex geometries.

The success rates of the developed method are consistently higher for the three linkage methods used, and the best performance of 81% is achieved for average linkage using the developed method. The increase in success rate is of at least a 13%, with the best alternative corresponding to the case of optimal choice of K using the descriptive function and complete linkage method. Our method requires no user input and outperforms the tested alternatives for all linkages, to the point that the worst performance of our method (72% for single linkage) is better than the best performance of the alternative methods (68%) for this application.

In addition to providing the best performance for our developed method, the average linkage is also the linkage method that results in the best average cophenetic coefficient for our dataset. This means it is the linkage method that better represents the route distributions in a dendrogram.

The approach presented in this work still fails in some scenarios with very short length scale, when local phenomena in the approach phases becomes geometrically dominant, but this scenarios are rare. It may also fail in some scenarios with extremely heavy density variations, usually combined with disperse clusters, although one may argue whether the clusters with low density are relevant in those situations. Ultimately, some scenarios have no clear patterns and clustering may not be appropriate in those situations.

Overall, this method seems appropriate for the application of route clustering.

Chapter 12

Future Work

Some of the implicit problems of using the Fréchet distance as distance metric can be solved by using a variant of the Fréchet distance. As proposed in [12] Section 4, an alternative is to use the minimum of the total distance of an order-preserving correspondence between the points of a pair of curves. This approach would differentiate a pair of curves that are locally different from a pair of curves that are different for most of their lengths, whereas both pairs may have the same Fréchet distance. This is a promising distance metric which has not been tested.

It would be recommendable to test the behavior of an alternative quality measure. One would first calculate the average Silhouette score for each individual cluster, finally obtaining an overall average of all cluster averages. This approach, conceptually similar to the Davies-Bouldin index, would give the same importance to the quality of all clusters independently of their density. Only preliminary tests have been done, and further study of this alternative (or the Davies-Bouldin index itself) should be carried out.

The aggregate network resulting of this work must be used in real flow optimization problems in order to detect the existence of errors that are not strictly clustering-related. Work towards implementing the air traffic flow optimization problem is in process.

Bibliography

[1] FAA. Traffic flow management in the national air space system. http://www.fly.faa.gov/Products/Training/Traffic_Management_for_Pilots/TFM_in_the_NAS_Booklet_ca10.pdf, July 2015.

[2] FAA. Flight schedule monitor (fsm). http://cdm.fly.faa.gov/?page_id=174, July 2015.

[3] M. Greene, G. Pierce, R. Rosen, and J. Cistone. Assessment and alignment of nasa atm-airspace project with nextgen r&d needs. July 2008.

[4] Alessandro Bombelli, Lluis Soler, and Kenneth D. Mease. *Strategic Air Traffic Planning with Frechet Distance Aggregation and Rerouting*. American Institute of Aeronautics and Astronautics, August 2015.

[5] M Callaham, J DeArmon, A Cooper, J Goodfriend, Debra Moch-Mooney, and G Solomos. Assessing nas performance: Normalizing for the effects of weather. In *4th USA/Europe Air Traffic Management R&D Symposium*, 2001.

[6] Banavar Sridhar and Neil Chen. Short-term national airspace system delay prediction using weather impacted traffic index. *Journal of guidance, control, and dynamics*, 32(2):657–662, 2009.

[7] FAA. Aspm airports. http://aspmhelp.faa.gov/index.php/ASPM_Airports, November 2015.

[8] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[9] Lovisa Lovmar, Annika Ahlford, Mats Jonsson, and Ann-Christine Syvänen. Silhouette scores for assessment of snp genotype clusters. *BMC genomics*, 6(1):35, 2005.

[10] Periklis Andritsos. Data clustering techniques. *Rapport technique, University of Toronto. Department of Computer Science*, 2002.

[11] M Maurice Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[12] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. Technical report, Citeseer, 1994.

[13] Saikat Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE, 1999.

[14] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM, 1998.

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

[16] George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[17] Yu Qian, Gang Zhang, and Kang Zhang. Façade: a fast and effective approach to the discovery of dense clusters in noisy spatial data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 921–922. ACM, 2004.

[18] JAS Almeida, LMS Barbosa, AACC Pais, and SJ Formosinho. Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering. *Chemometrics and Intelligent Laboratory Systems*, 87(2):208–217, 2007.

[19] Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463), 2003.

[20] Stan Salvador and Philip Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 576–584. IEEE, 2004.

[21] Jörg Sander, Xuejie Qin, Zhiyong Lu, Nan Niu, and Alex Kovarsky. Automatic extraction of clusters from hierarchical clustering representations. In *Advances in knowledge discovery and data mining*, pages 75–87. Springer, 2003.

[22] Hans-Peter Kriegel, Stefan Brecheisen, Eshref Januzaj, Peer Kröger, and Martin Pfeifle. Visual mining of cluster hierarchies. In *Proceedings 3rd International Workshop on Visual Data Mining (VDM@ ICDM2003)*, pages 151–165. Citeseer, 2003.

[23] Stefan Brecheisen, Hans-Peter Kriegel, Peer Kröger, and Martin Pfeifle. Visually mining through cluster hierarchies. In *SDM*, pages 400–411. SIAM, 2004.

[24] M Daszykowski, B Walczak, and DL Massart. Looking for natural patterns in data: Part 1. density-based approach. *Chemometrics and Intelligent Laboratory Systems*, 56(2):83–92, 2001.

[25] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod Record*, volume 28, pages 49–60. ACM, 1999.

[26] Michael Daszykowski, Beata Walczak, and Desire L Massart. Looking for natural patterns in analytical data. 2. tracing local density with optics. *Journal of chemical information and computer sciences*, 42(3):500–507, 2002.

[27] F. James Rohlf Robert R. Sokal. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40, 1962.

[28] John A Hartigan and PM Hartigan. The dip test of unimodality. *The Annals of Statistics*, pages 70–84, 1985.

Appendix A

Additional Examples

Several subsets of different nature will be included now, with the plots corresponding to the different steps in the clustering process, to show the behavior of the developed method.

All the presented results are produced in a fully automatic way, without any interference from the user at any point.

Memphis International Airport: big density variations between clusters.

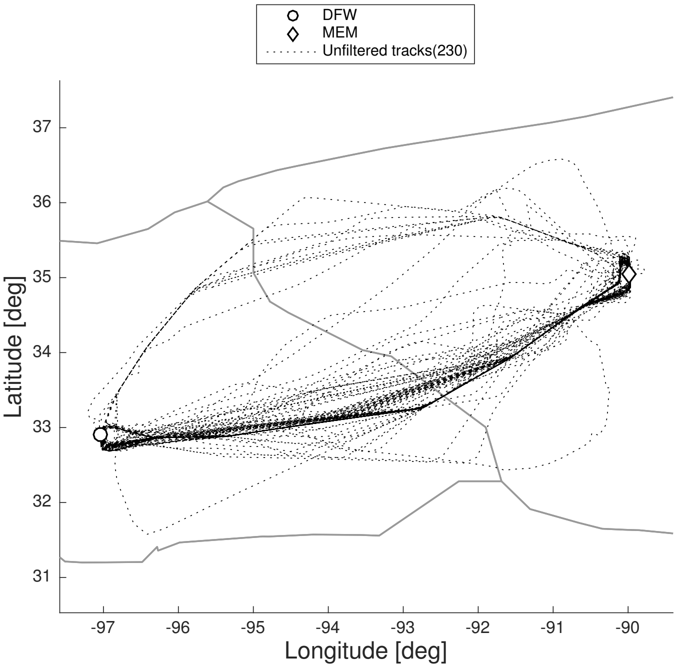

Figure A.1: Unfiltered tracks, MEM.

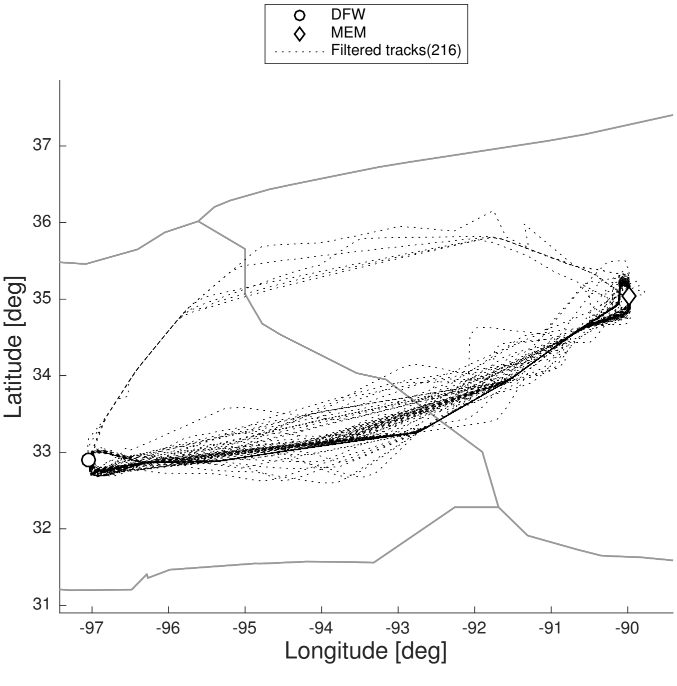

Figure A.2: Tracks without outliers, MEM.

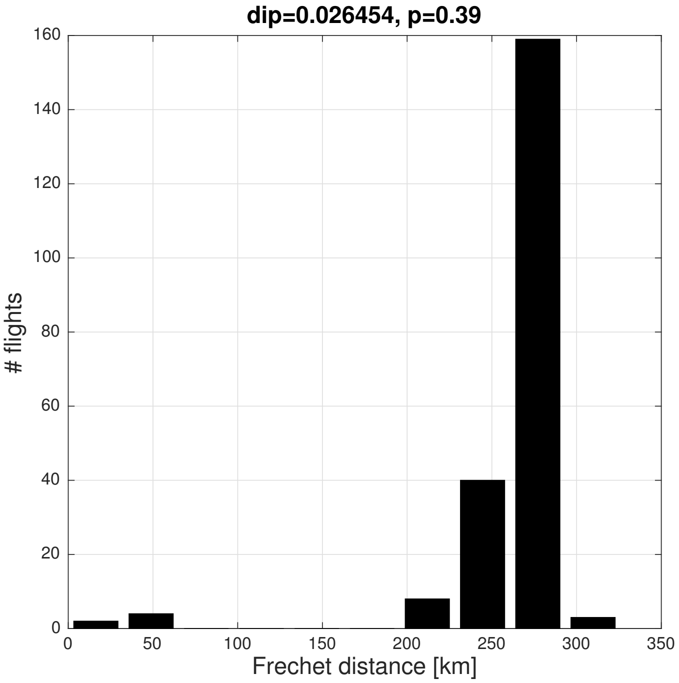

Figure A.3: Dip test, MEM.

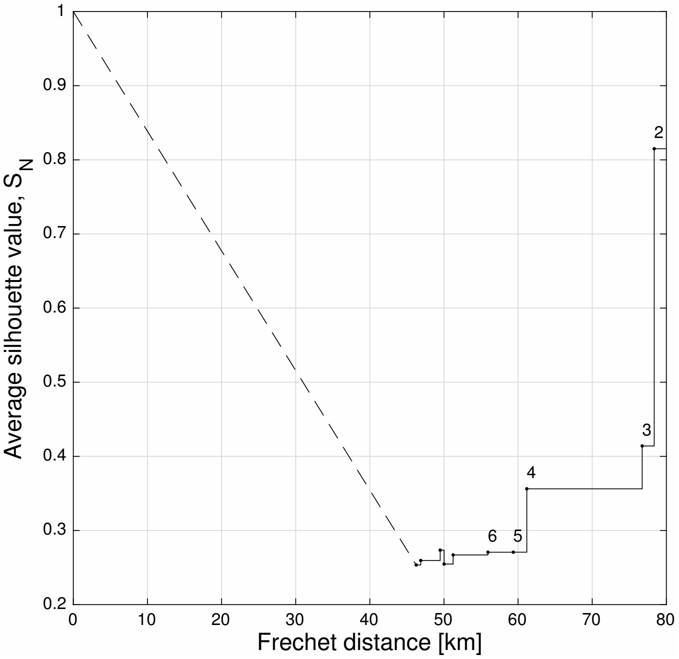

Figure A.4: $\bar{S}_N$ evolution, MEM.

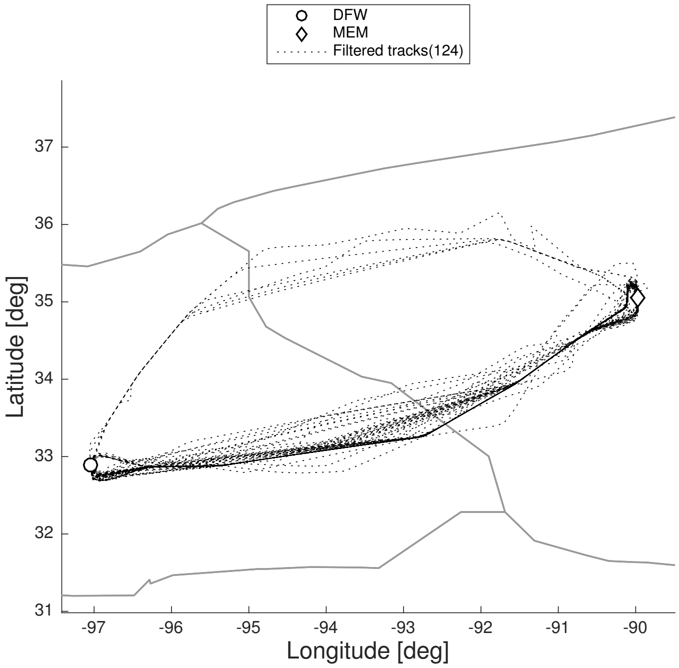

Figure A.5: Tracks without outliers after speed filtering, MEM.

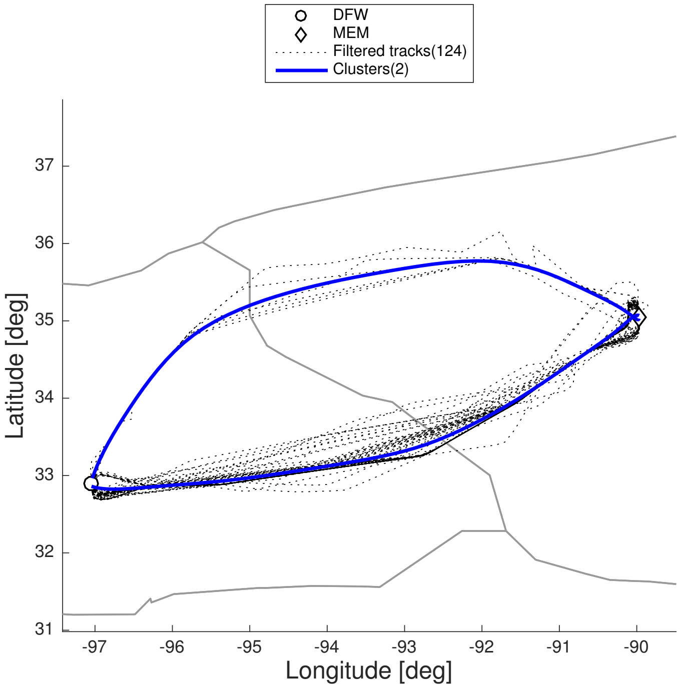

Figure A.6: Final tracks and clusters, MEM.

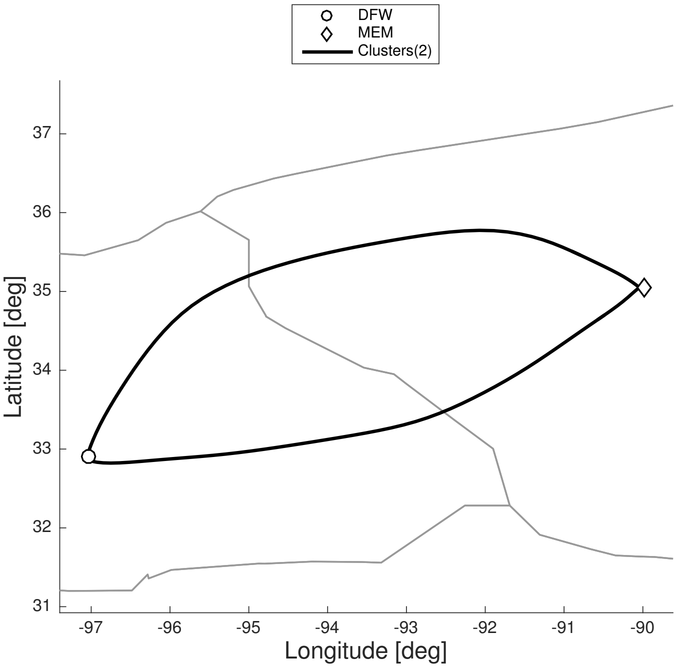

Figure A.7: Clusters, MEM

Southwest Florida International Airport: big density variations and disperse clusters.

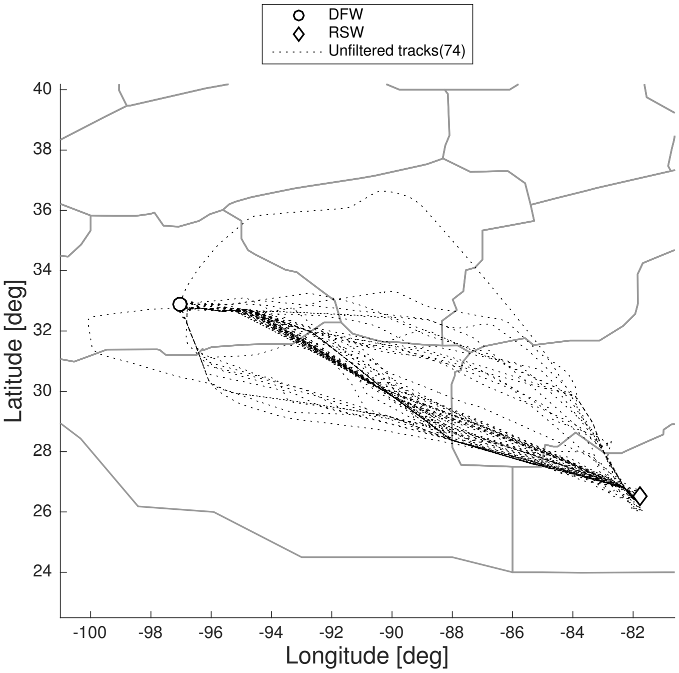

Figure A.8: Unfiltered tracks, RSW.

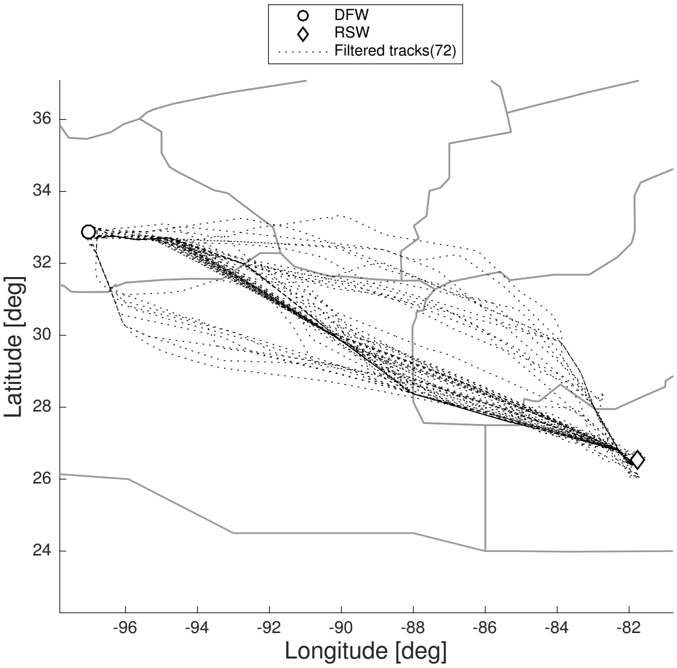

Figure A.9: Tracks without outliers, RSW.

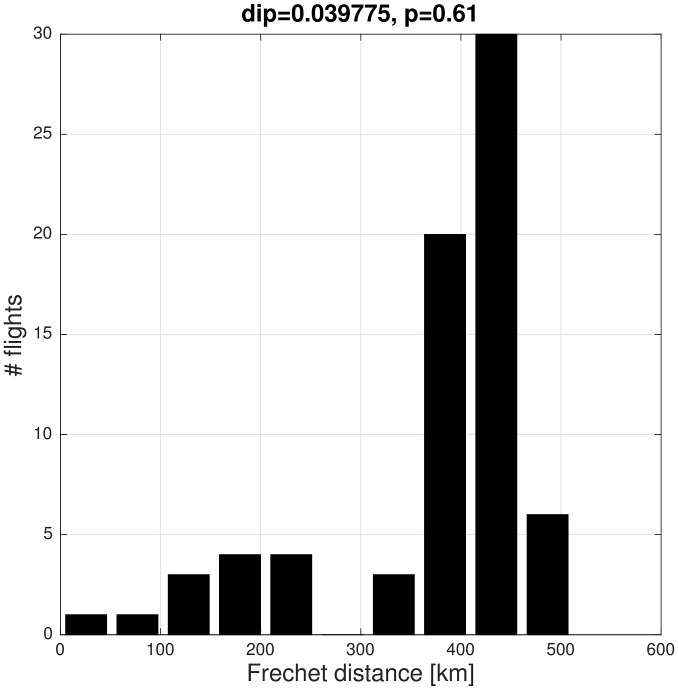

Figure A.10: Dip test, RSW.

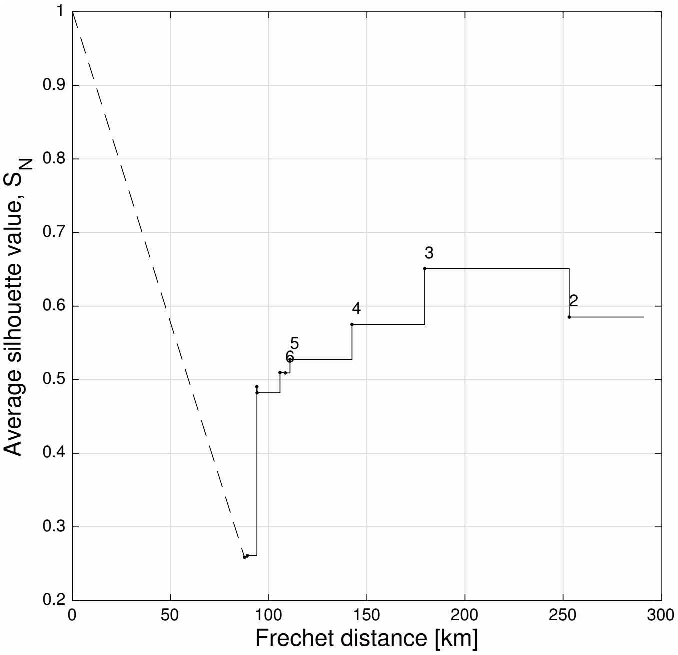

Figure A.11: $\bar{S}_N$ evolution, RSW.

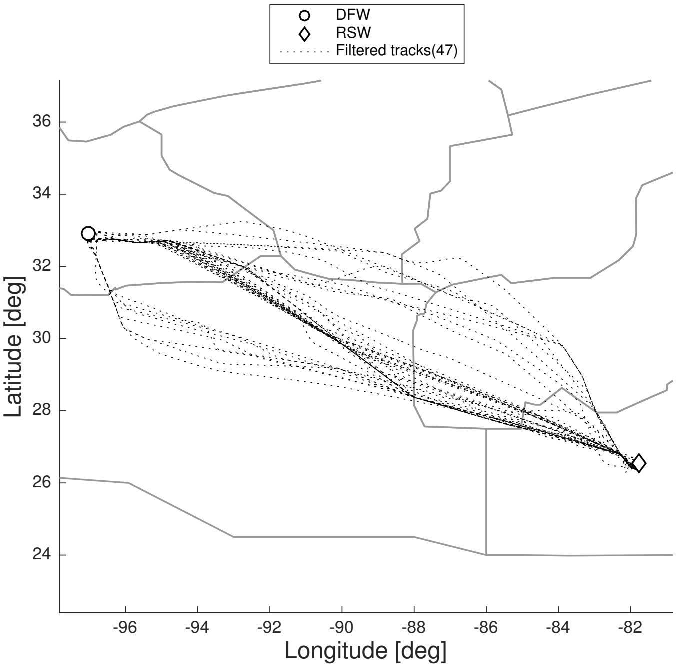

Figure A.12: Tracks without outliers after speed filtering, RSW.

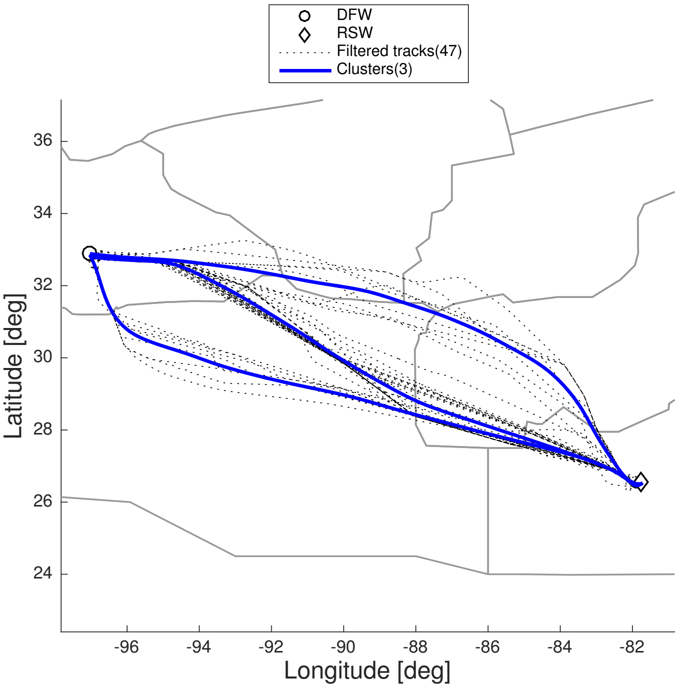

Figure A.13: Final tracks and clusters, RSW.

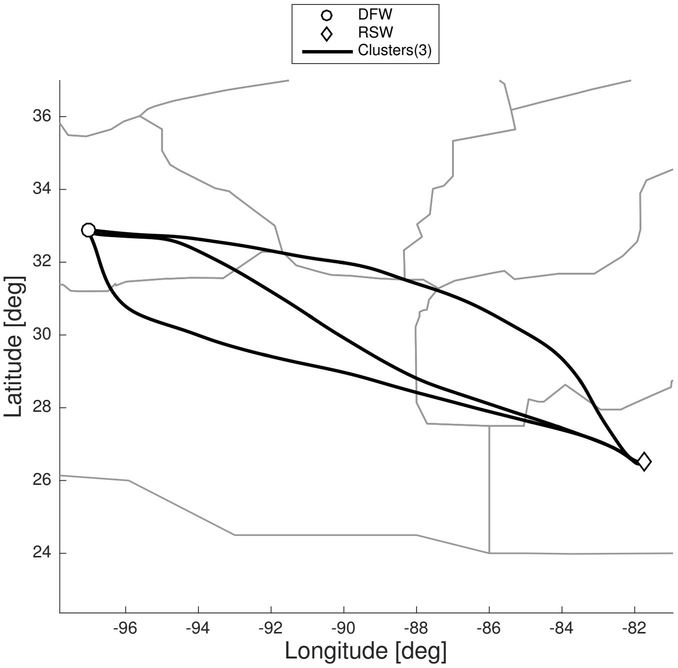

Figure A.14: Clusters, RSW

General Mitchell International Airport: big density variations and disperse clusters.

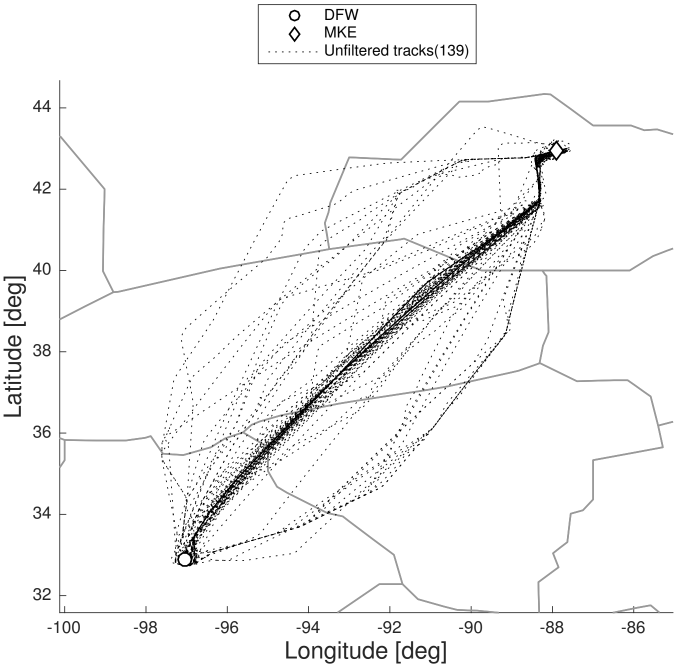

Figure A.15: Unfiltered tracks, MKE.

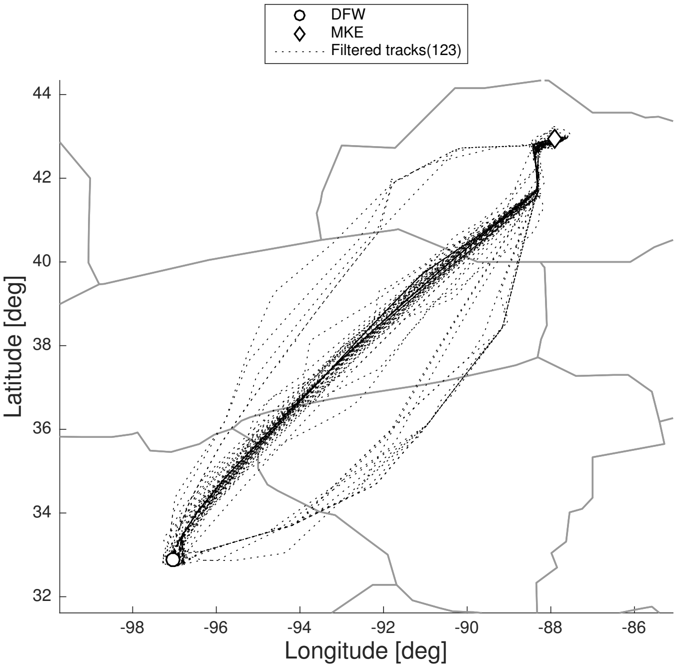

Figure A.16: Tracks without outliers, MKE.

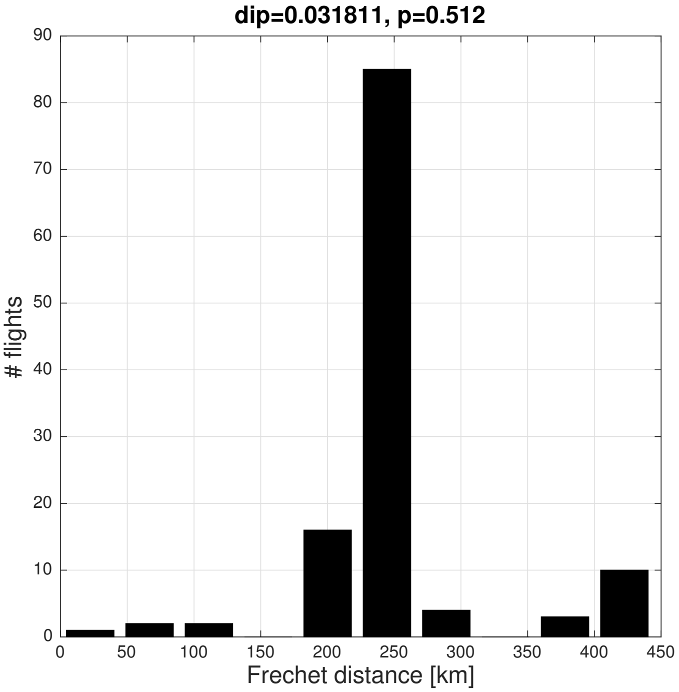

Figure A.17: Dip test, MKE.

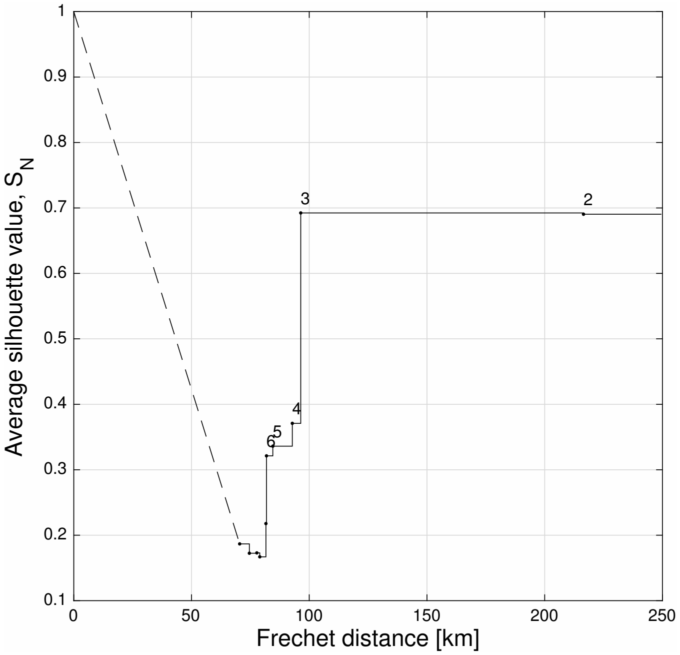

Figure A.18: $\bar{S}_N$ evolution, MKE.

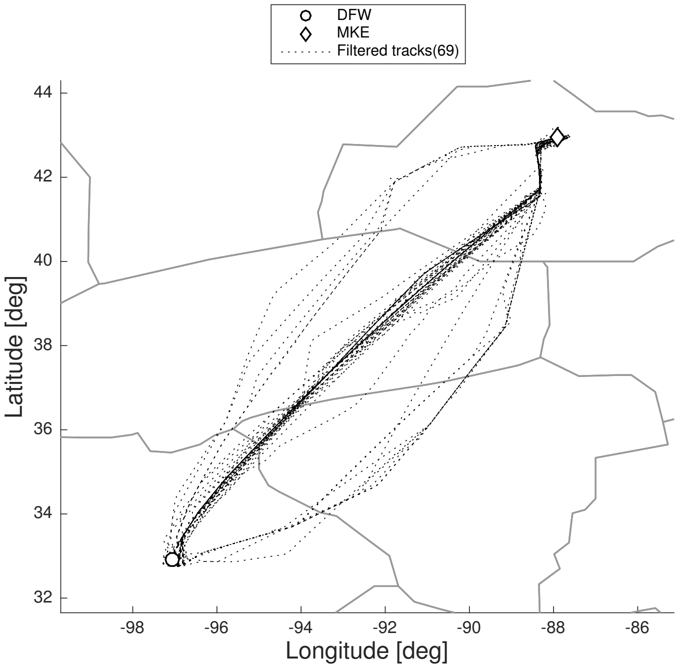

Figure A.19: Tracks without outliers after speed filtering, MKE.

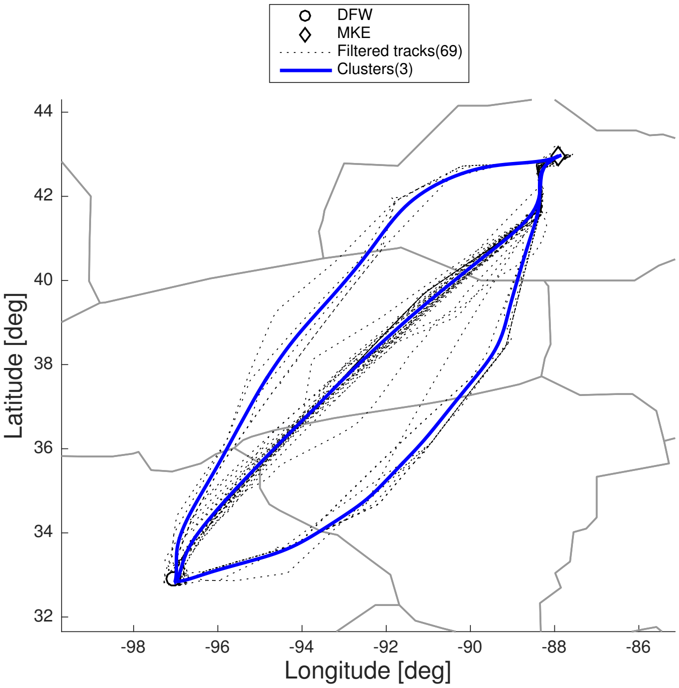

Figure A.20: Final tracks and clusters, MKE.

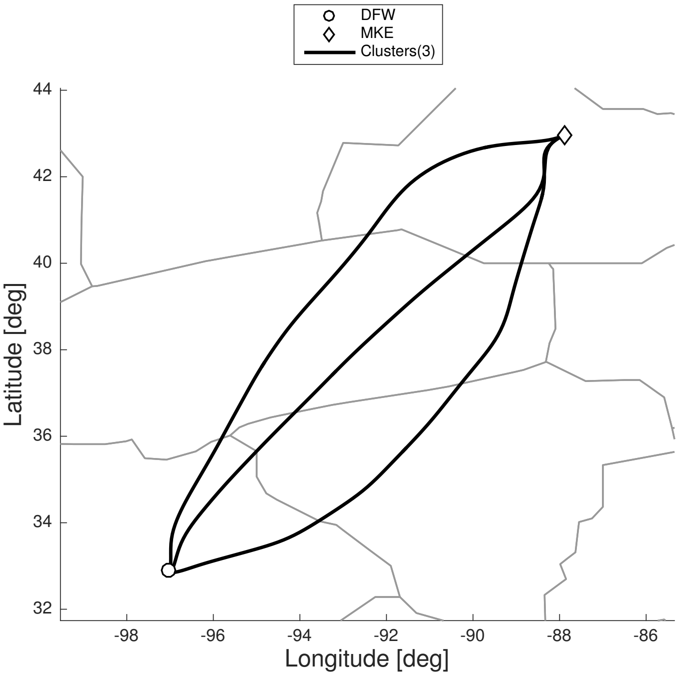

Figure A.21: Clusters, MKE

LaGuardia Airport: complex geometry, internal sub-cluster structures and many outliers.

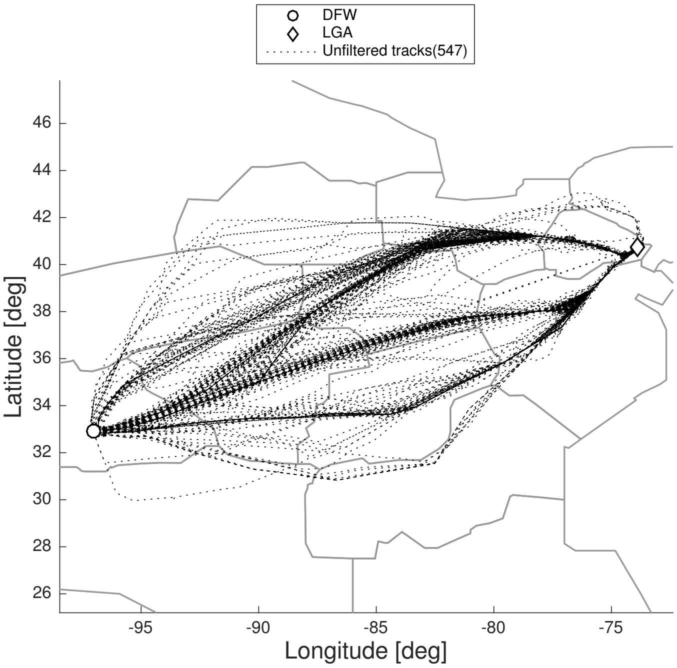

Figure A.22: Unfiltered tracks, LGA.

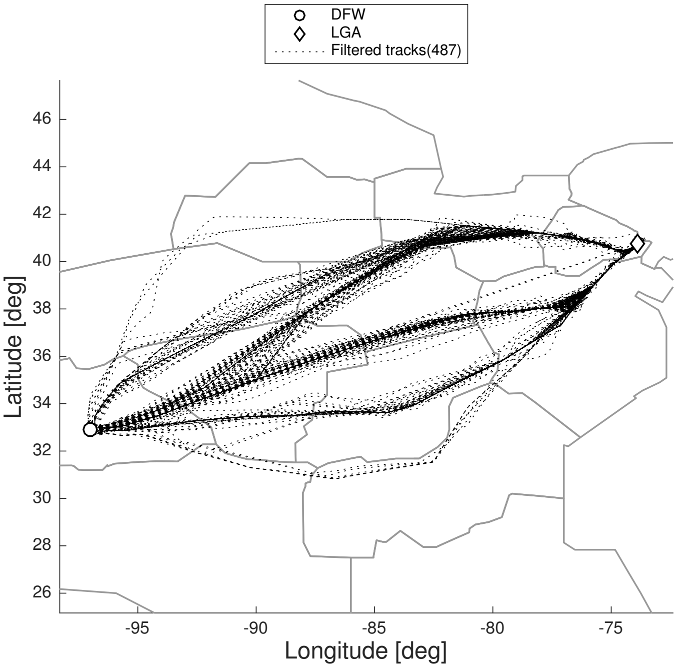

Figure A.23: Tracks without outliers, LGA.

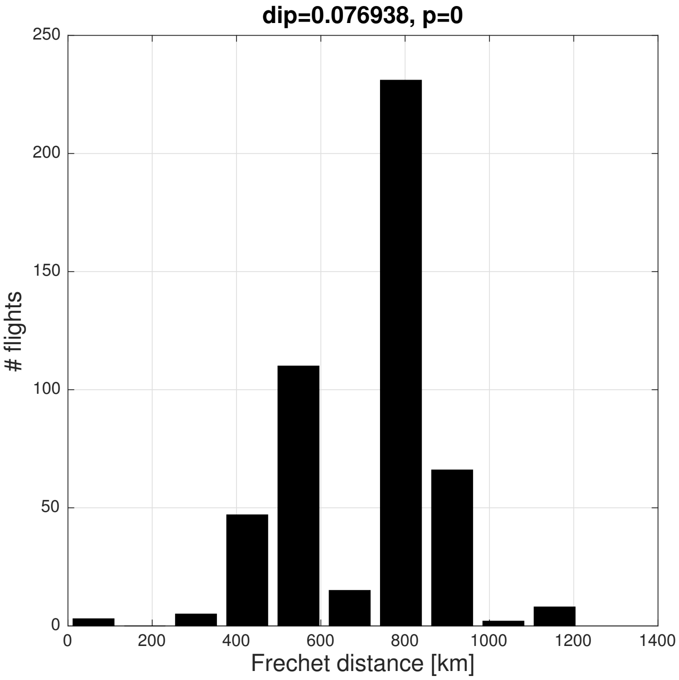

Figure A.24: Dip test, LGA.

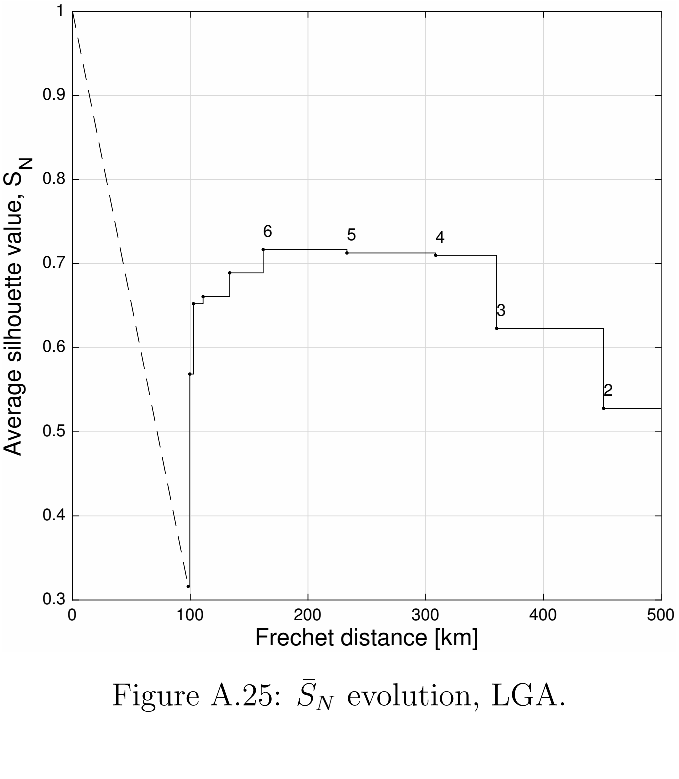

Figure A.25: $\bar{S}_N$ evolution, LGA.

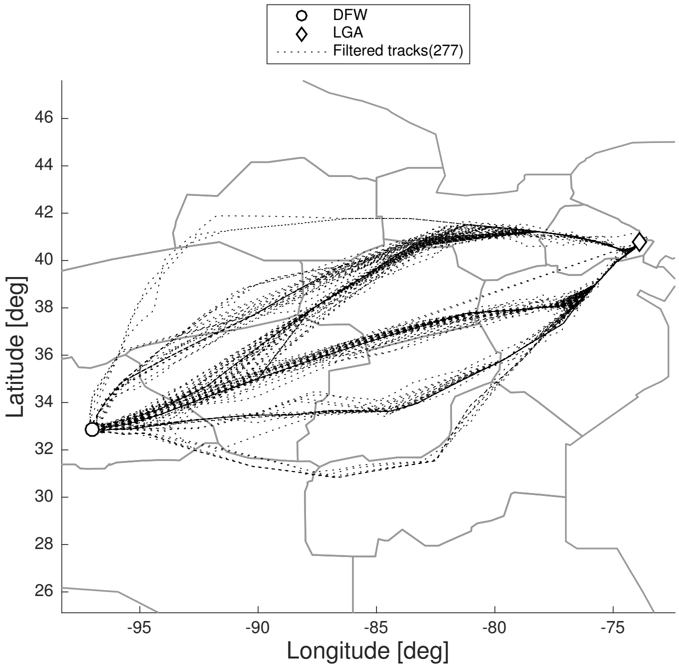

Figure A.26: Tracks without outliers after speed filtering, LGA.

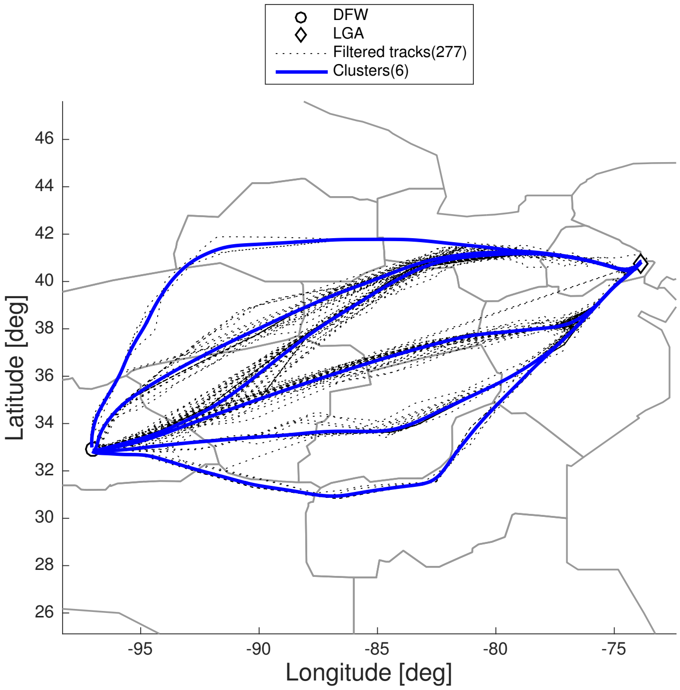

Figure A.27: Final tracks and clusters, LGA.

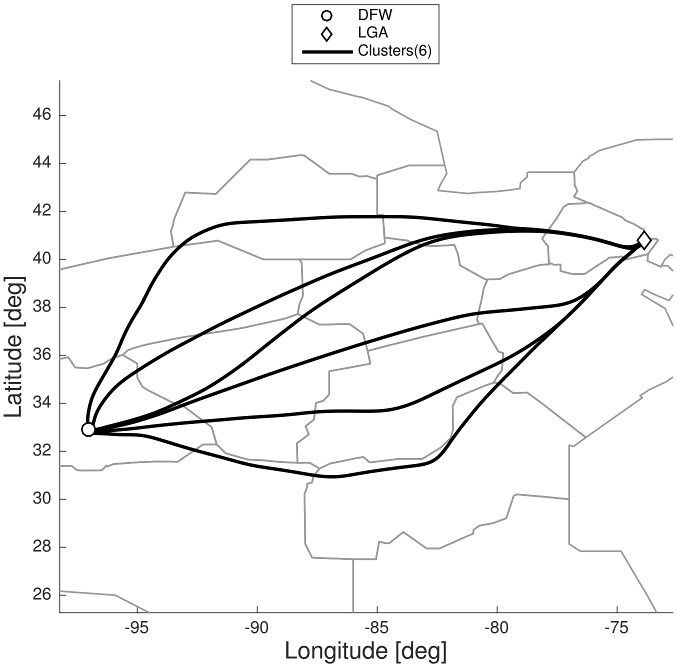

Figure A.28: Clusters, LGA

Sacramento International Airport: unimodal distribution.

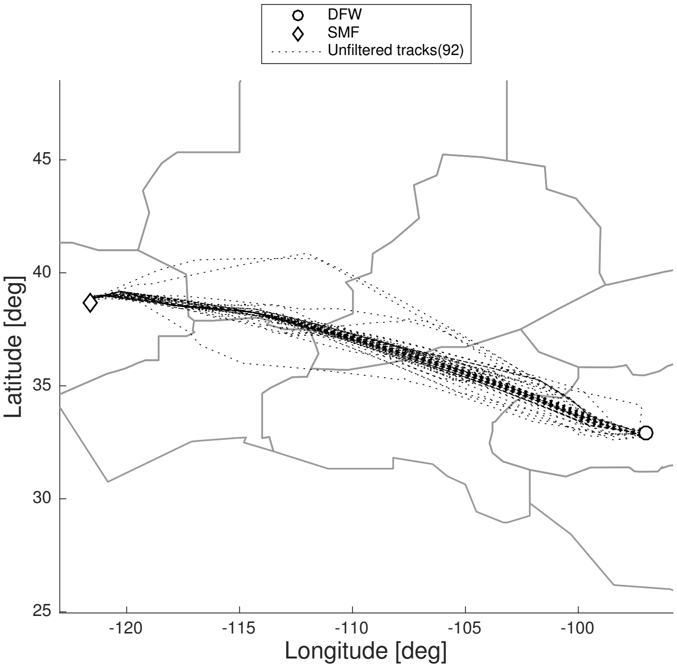

Figure A.29: Unfiltered tracks, SMF.

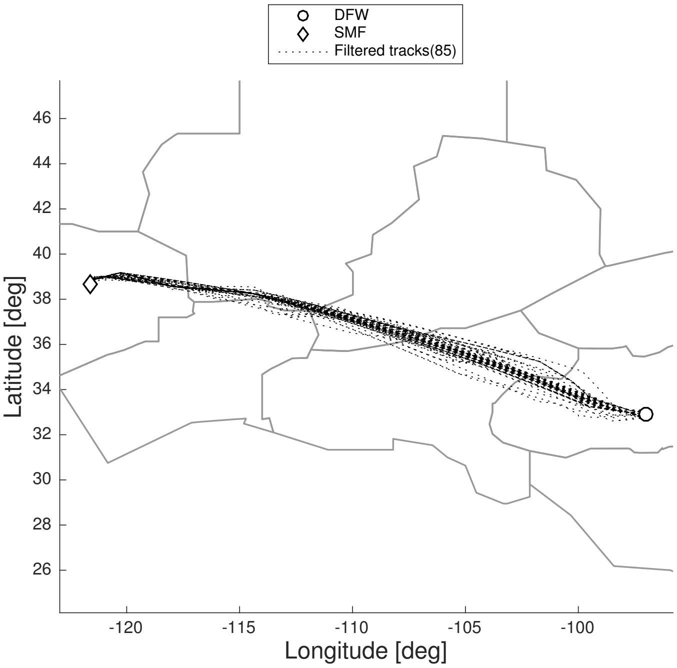

Figure A.30: Tracks without outliers, SMF.

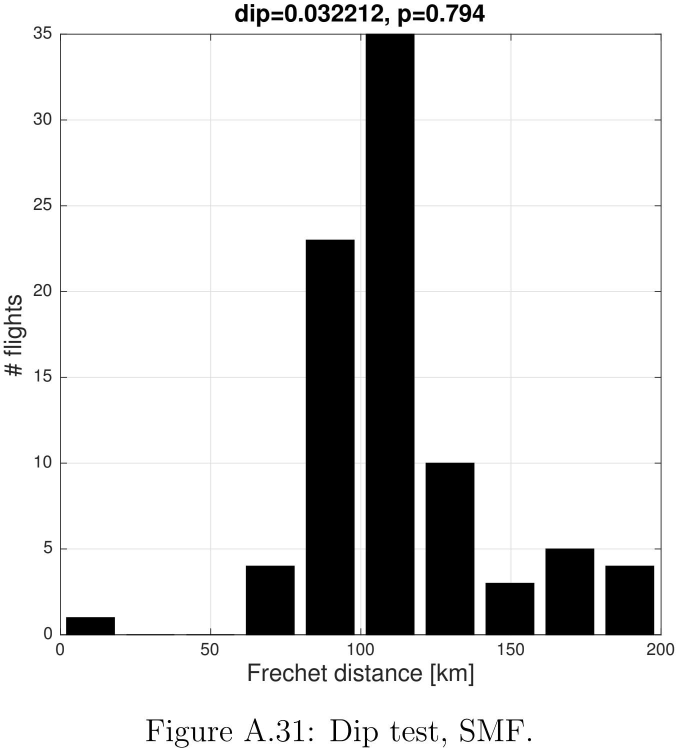

Figure A.31: Dip test, SMF.

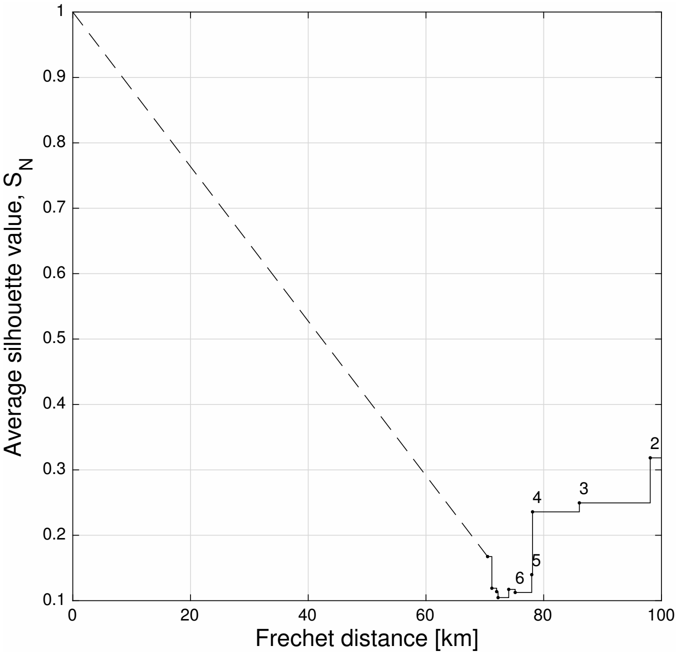

Figure A.32: $\bar{S}_N$ evolution, SMF.

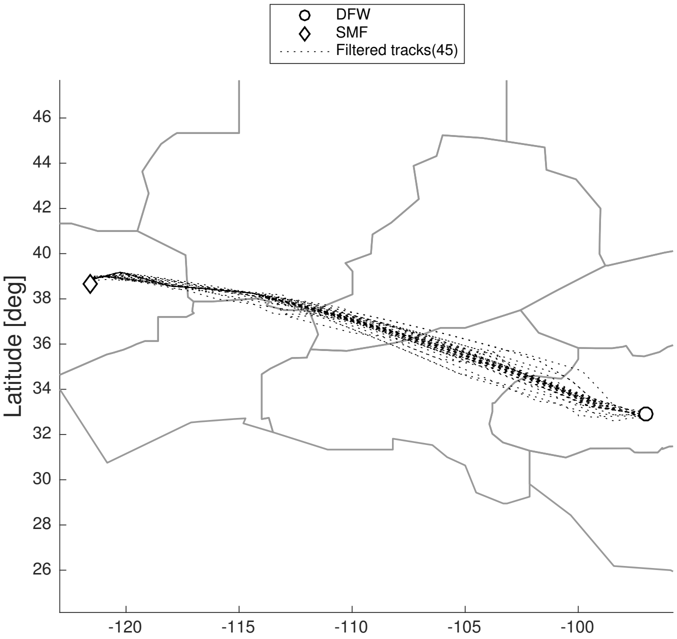

Figure A.33: Tracks without outliers after speed filtering, SMF.

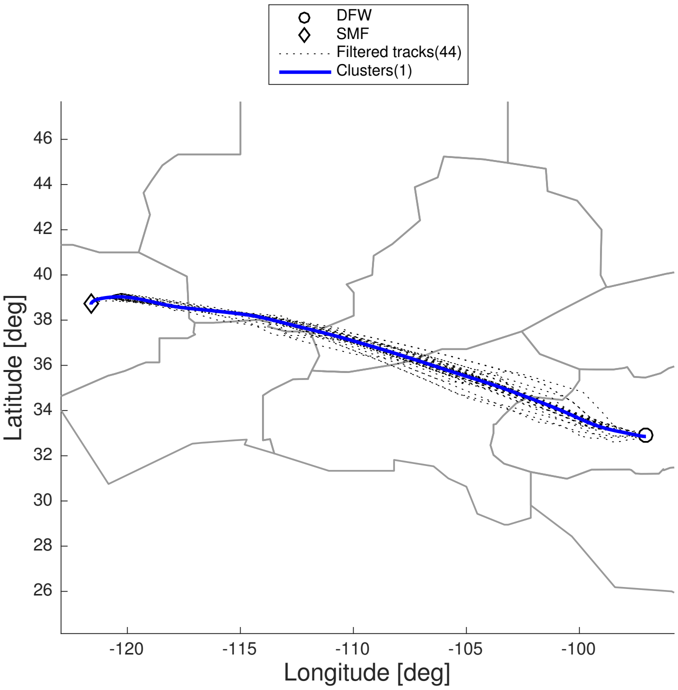

Figure A.34: Final tracks and clusters, SMF.

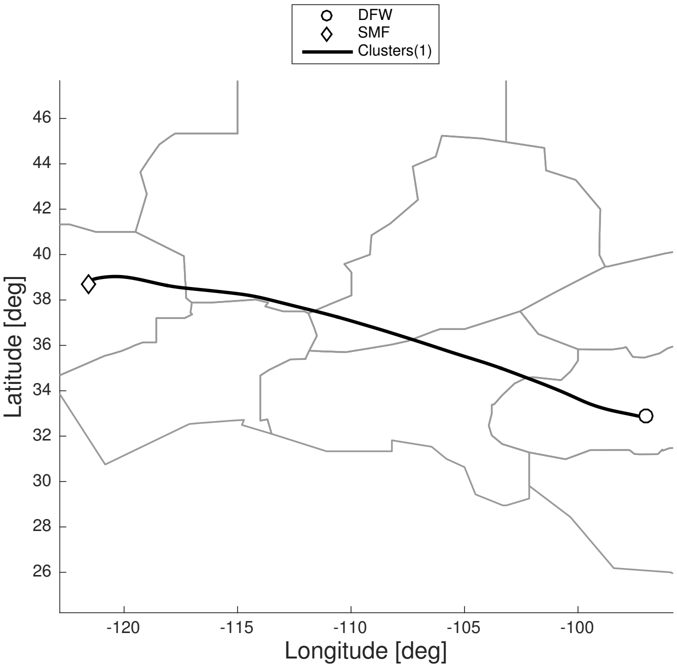

Figure A.35: Clusters, SMF

Los Angeles International Airport: very disperse clusters.

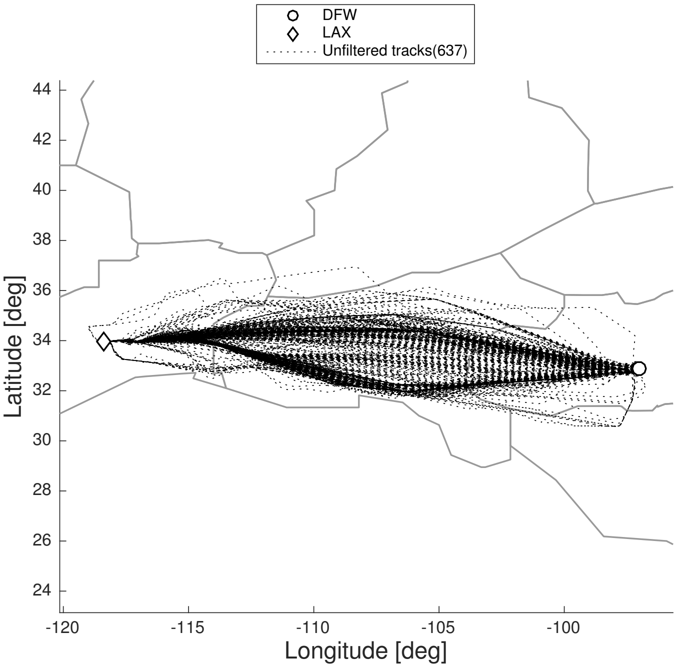

Figure A.36: Unfiltered tracks, LAX.

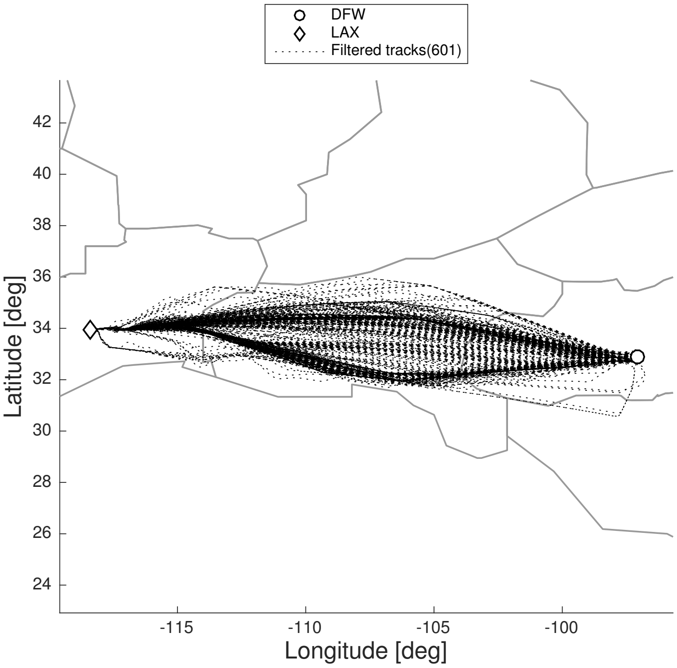

Figure A.37: Tracks without outliers, LAX.

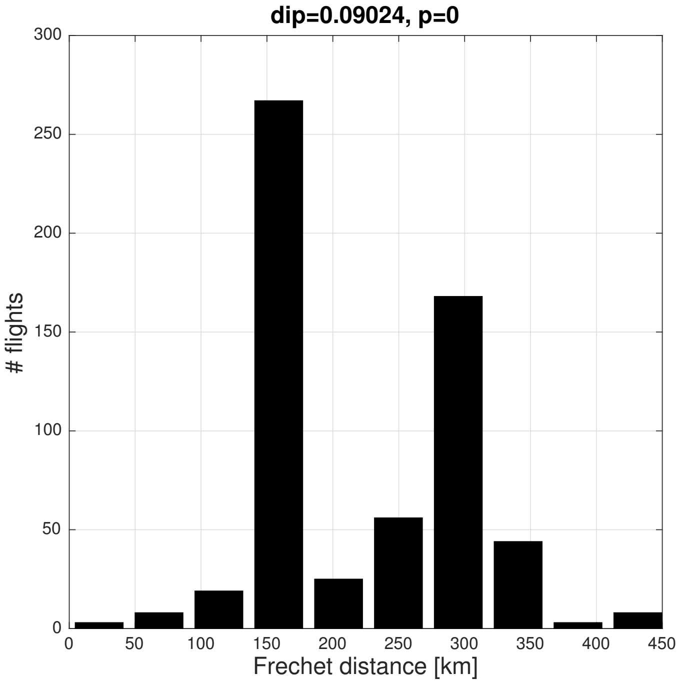

Figure A.38: Dip test, LAX.

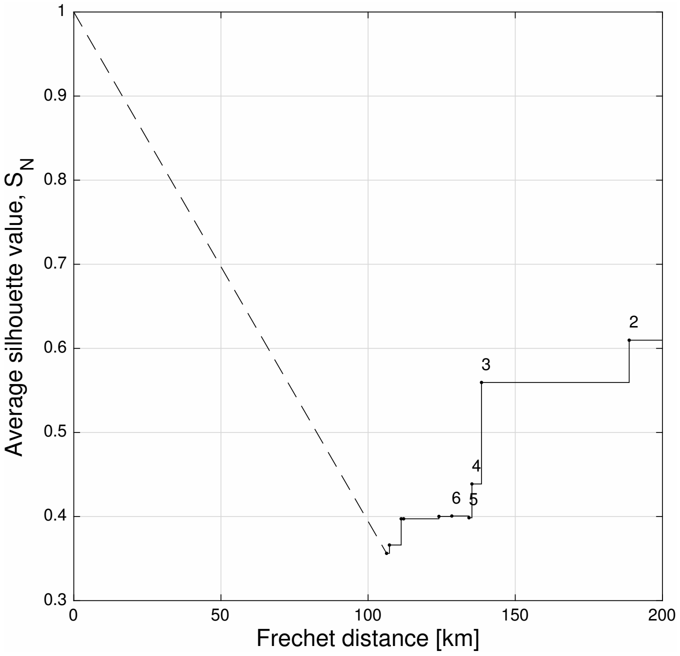

Figure A.39: $\bar{S}_N$ evolution, LAX.

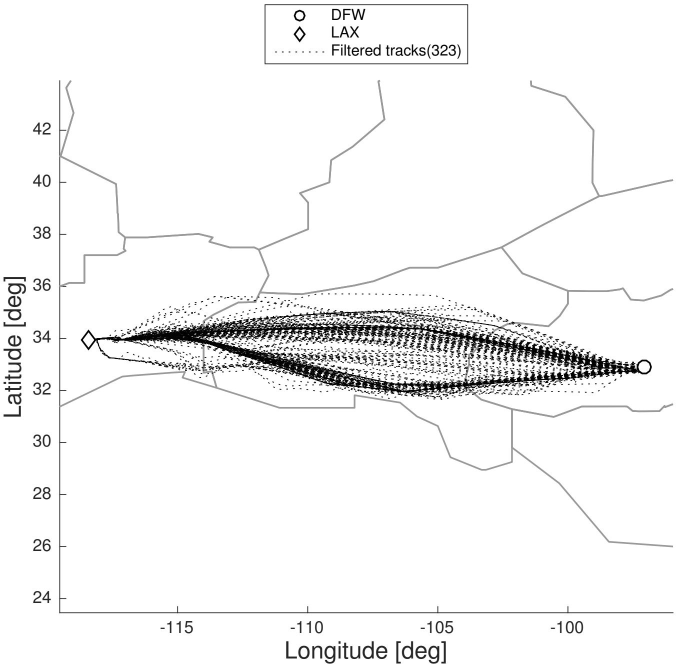

Figure A.40: Tracks without outliers after speed filtering, LAX.

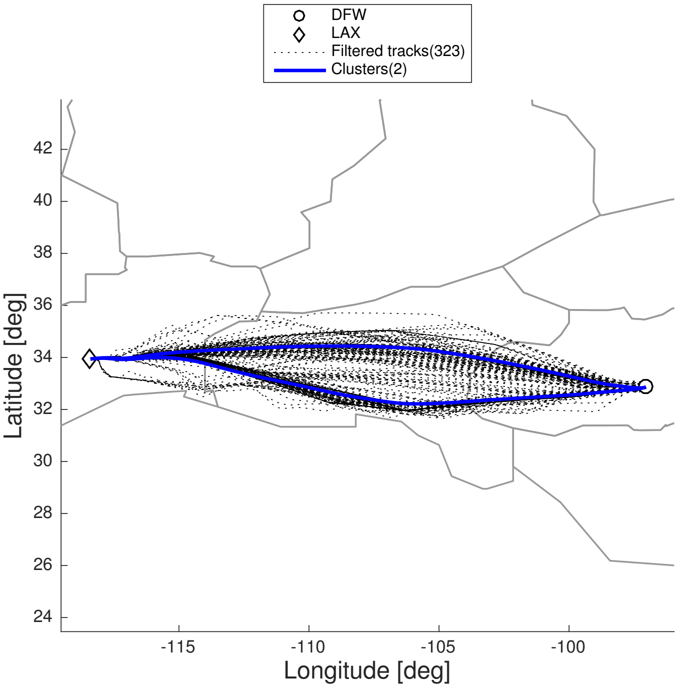

Figure A.41: Final tracks and clusters, LAX.

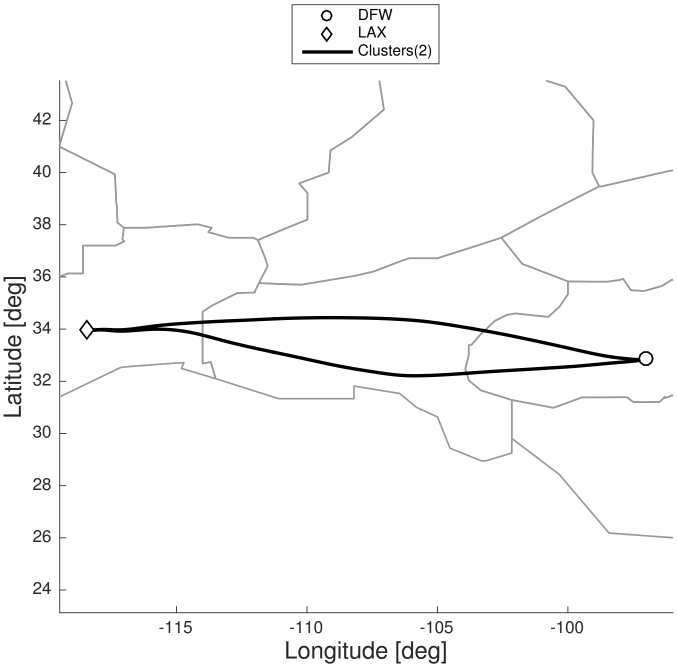

Figure A.42: Clusters, LAX

Palm Springs International Airport: disperse very low density clusters.

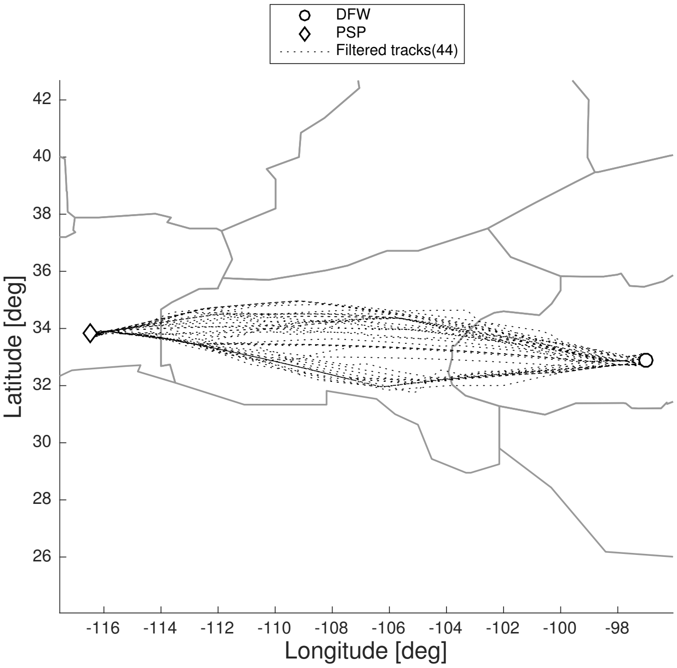

Figure A.43: Unfiltered tracks, PSP.

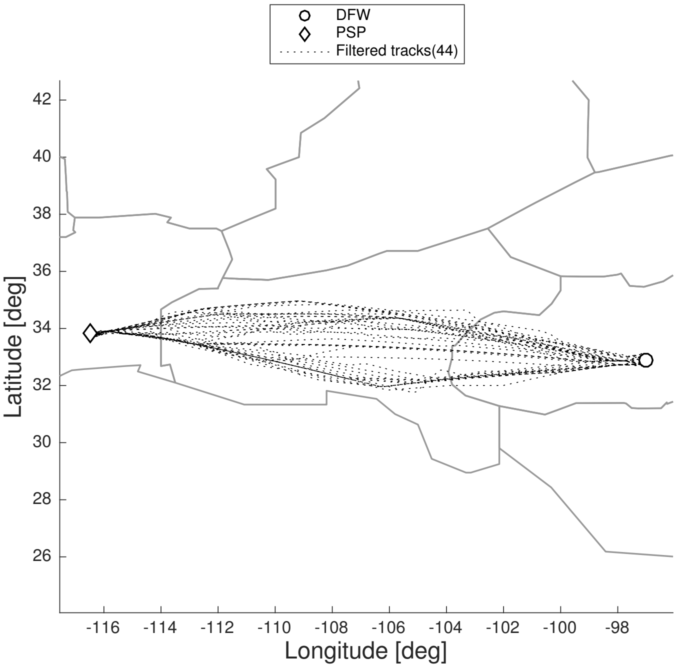

Figure A.44: Tracks without outliers, PSP.

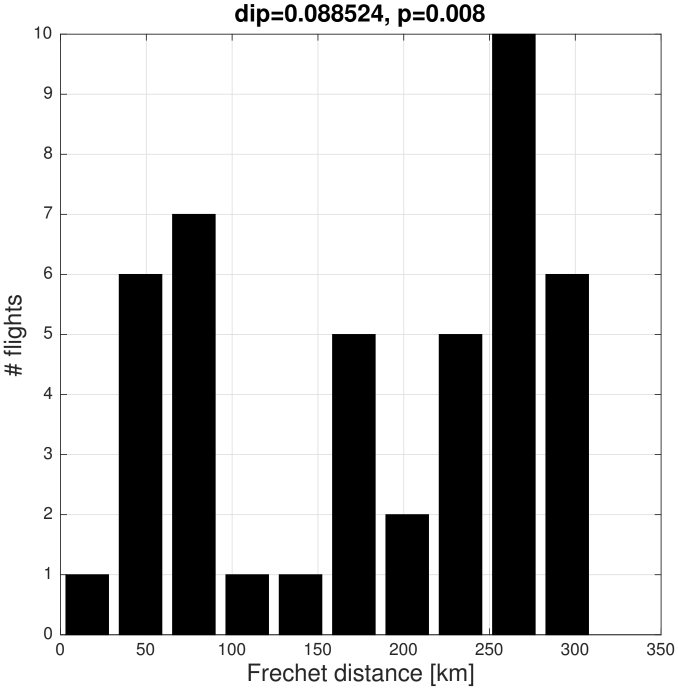

Figure A.45: Dip test, PSP.

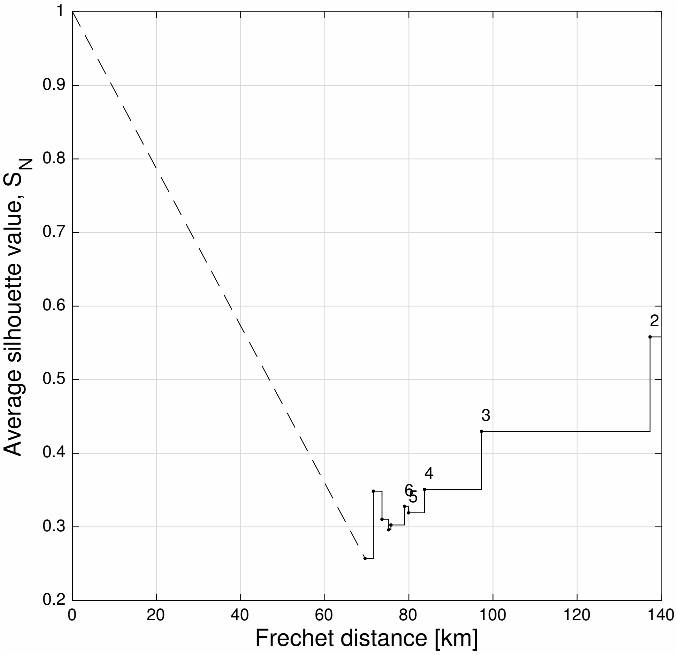

Figure A.46: $\bar{S}_N$ evolution, PSP.

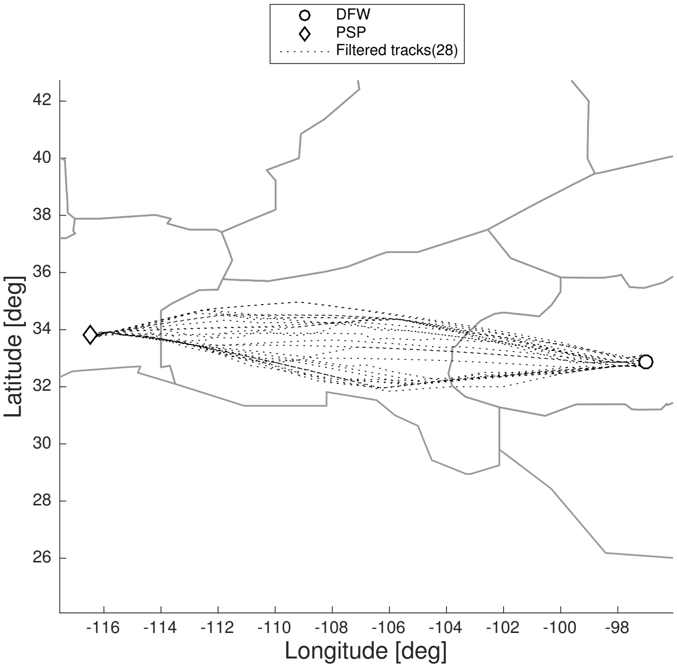

Figure A.47: Tracks without outliers after speed filtering, PSP.

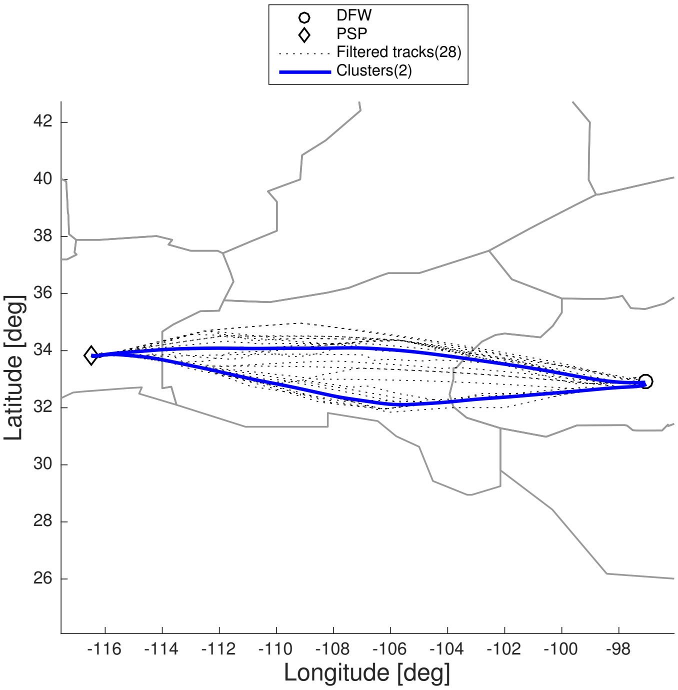

Figure A.48: Final tracks and clusters, PSP.

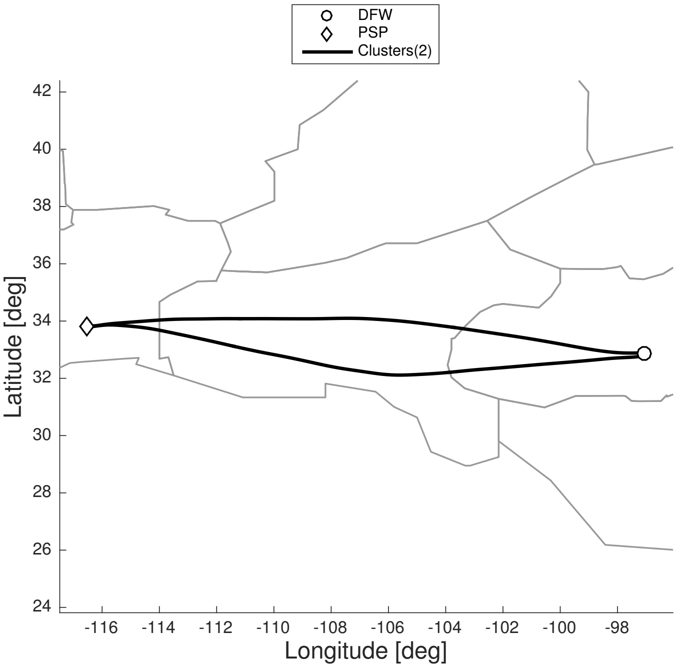

Figure A.49: Clusters, PSP

Hartsfield-Jackson Atlanta International Airport: density variations.

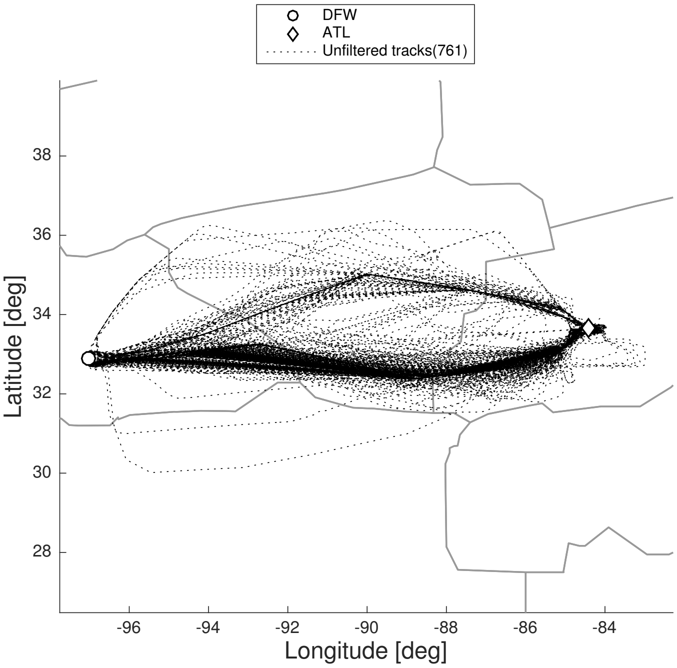

Figure A.50: Unfiltered tracks, ATL.

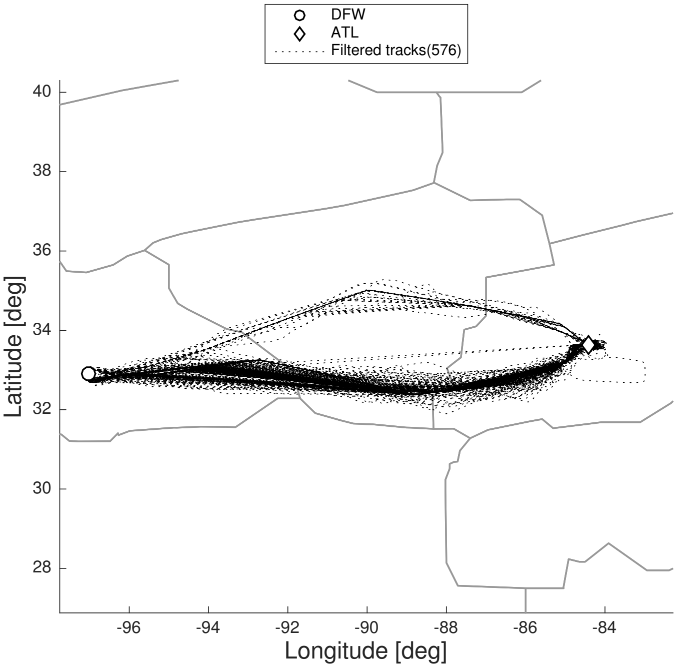

Figure A.51: Tracks without outliers, ATL.

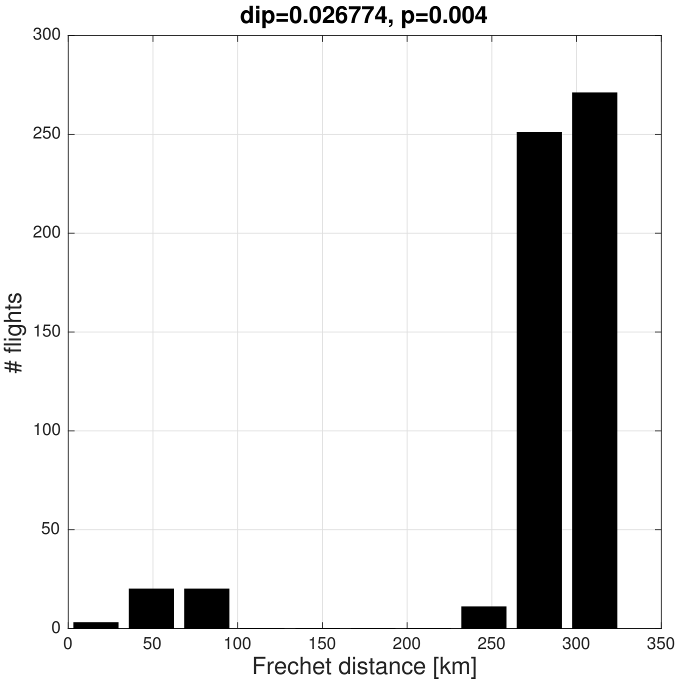

Figure A.52: Dip test, ATL.

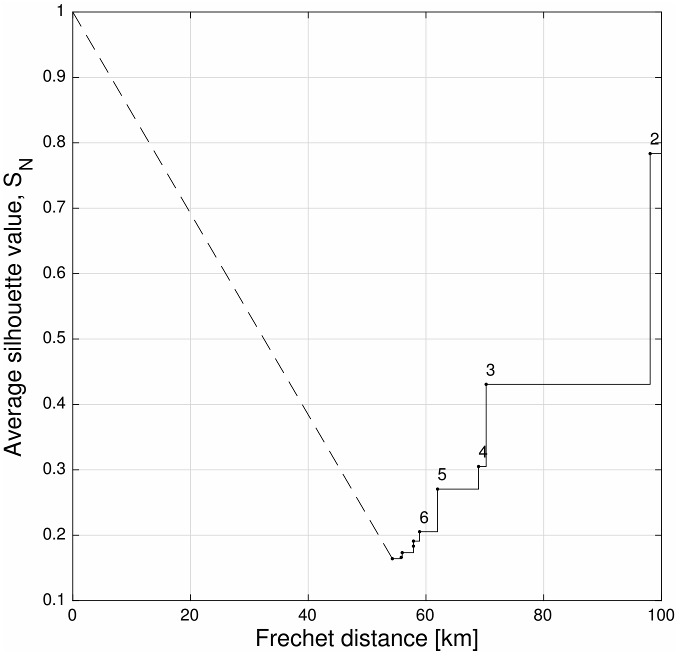

Figure A.53: $\bar{S}_N$ evolution, ATL.

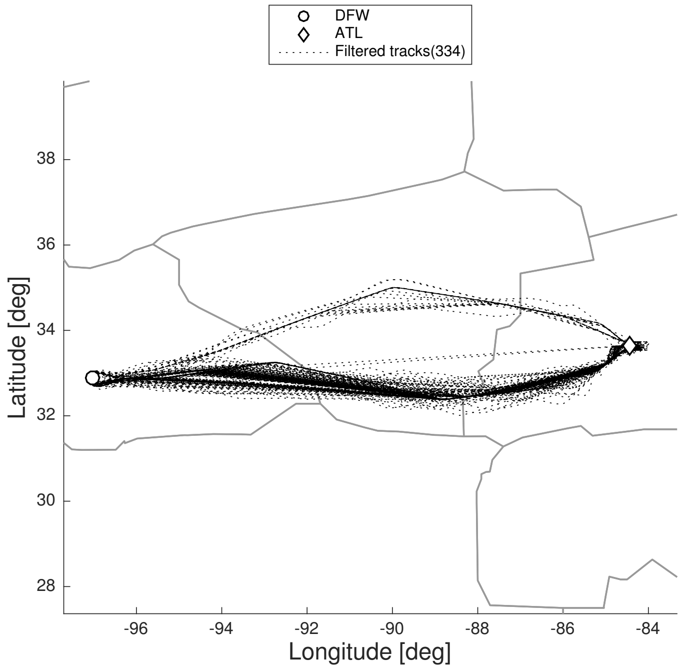

Figure A.54: Tracks without outliers after speed filtering, ATL.

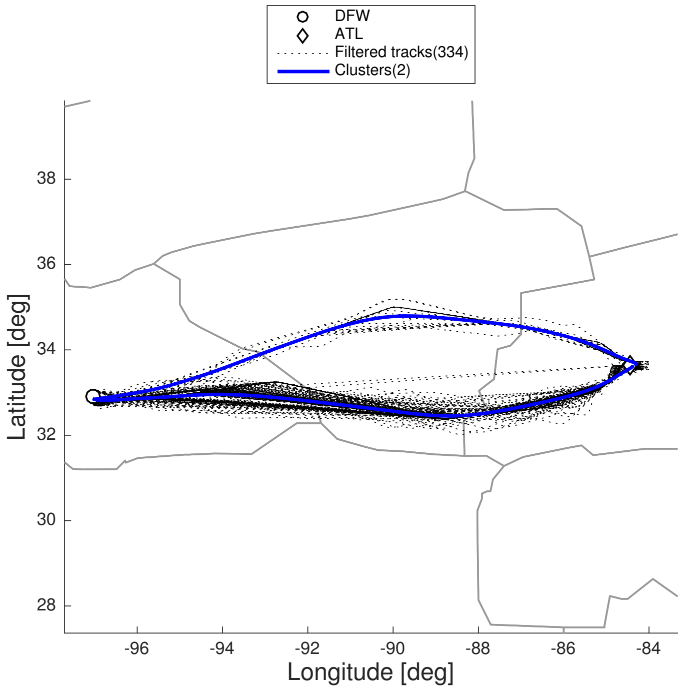

Figure A.55: Final tracks and clusters, ATL.

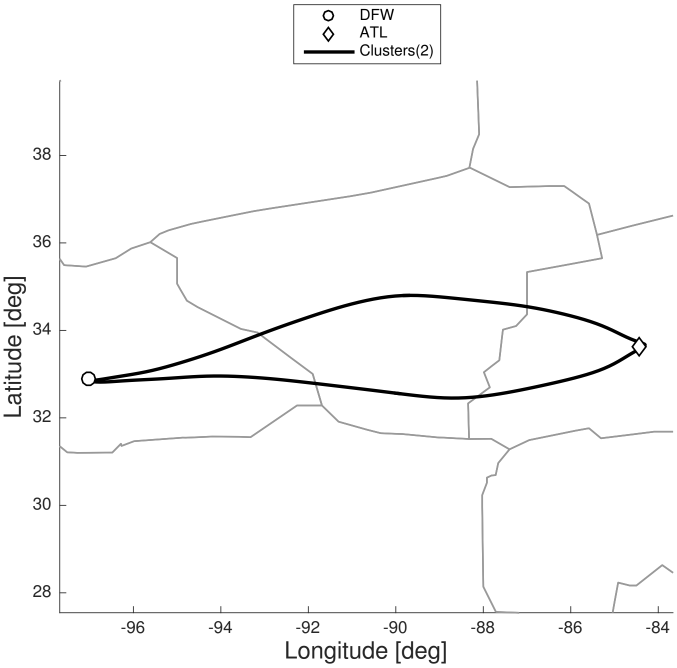

Figure A.56: Clusters, ATL

In the shown examples there is cases with big density variations between clusters, unimodal distributions, complex patterns, internal sub-cluster structures, many outliers, disperse clusters and low density clusters. The developed clustering method provides successful results in a wide variety of cases.