Documentation MR1Analysis Tool

Project:	MR1Analysis Tool
Contractor:	Swiss Birdradar Solution AG
Author:	Fabian Hertner
Project Leader:	Dominik Kleger
Version:	1.0
Date:	08.04.21

Table of Content

1 Introduction

This document describes the use of the data produced by the BirdScan MR1 v1.6.0.7 and greater as well as the functionality of the MR1Analysis Tool. The MR1Analysis Tool is a collection of functions, written in “R”, to extract the data from the database and perform basic analysis such as MTR computation, wing-beat and direction analysis.

The MR1 Analysis Tool requires data collected by a BirdScan MR1 running BirdScan software version 1.6 or higher. Data collected by older BirdScan versions, reclassified with BirdScan v1.6 or higher is supported as well.

With the BirdScan v1.6 a new classifier based on the random forest algorithm was released. The previous naive bayes based classification is still computed and stored in the database as well as all the features.

2 MR1 Data

2.1 Database structure

The structure of the database (BirdScan v1.6) is shown in Figure 1. The figure shows the tables used by the MR1 Analysis tool. The database contains more tables than shown in Figure 1 which are not relevant for analysis purposes.

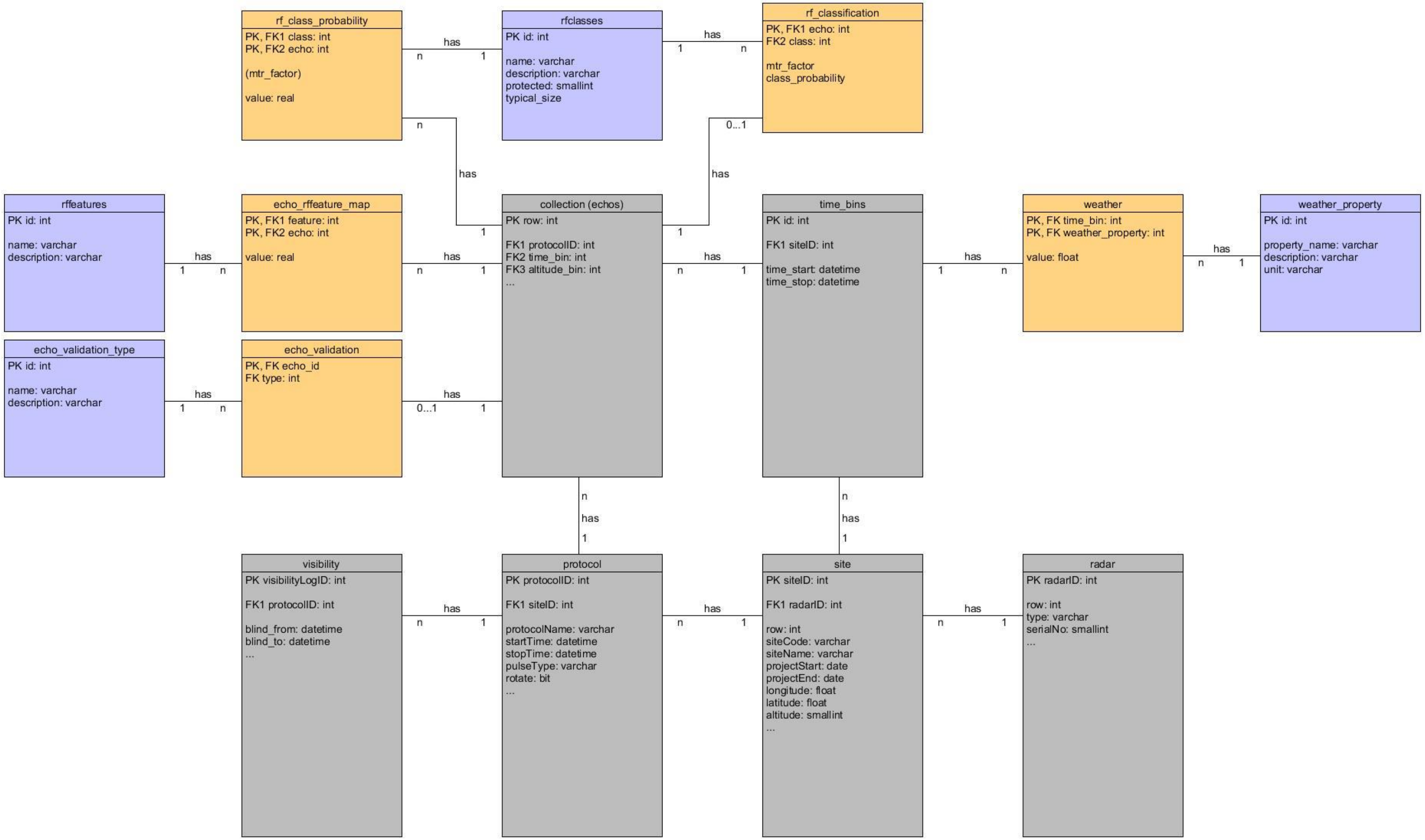

Figure 1: part of the database structure BirdScan v1.6

2.2 Database Tables

2.2.1 Collection

The collection table stores information about the detected echoes, one row per echo.

key type	name	datatype	linked to table
primary key	row	int	-
foreign key	protocolID	int	protocol
foreign key	time_bin	int	time_bins

Table 1: collection table keys

column	datatype	description	outdated by birdscan v1.6
echoID	int	Echo ID	no
protocolID	int	link to protocol table	no
stc_level	float	stc_level in dBm, used to calculate mtr-factors	no
mtr_fact	float	mtr-factor based on the old classification	yes
time_bin	int	link to time_bins table	no
statistical_classification	smallint	naive bayes classification	yes
time_stamp	datetime	timestamp of the echo	no
feature1	float	altitude in meters a.g.l.	no
feature2	float	azimuth in degree	no
feature3	float	speed in m/s	no
feature6	float	radar rotation frequency in Hz	no
feature14	float	maxlevel received in dBm	no
feature15	float	polarisation ratio	yes
feature16	float	absolute polarisation	yes
feature17	float	radar cross section (RCS) in m^2	no
feature18	float	square root of RCS	no
feature19	float	duration of echo in s	no
feature20	float	duration of echo in STC	no
feature24	float	alpha: angle between entry- and exit angle in the detection range in degrees	no

Table 2: important columns in collection table

2.2.2 time_bins

Every 5 minutes a new time bin is created and stored in the time_bins table. The time_bins table does not contain any information but start and stop time of the time bin. Other tables are linked to the time_bins table and store information per time bin or allow filtering by time bin.

key type	name	datatype	linked to table
primary key	id	int	-
foreign key	siteID	int	site

Table 3: time_bins table keys

column	datatype	description	outdated by birdscan v1.6
time_start	datetime	start time and date of the time bin	no
time_stop	datetime	stop time and date of the time bin	no

Table 4: important columns in time_bins table

2.2.3 protocol

The protocol table contains information about the operation mode of the radar. On each change of the operation mode (pulse-type, rotation, …) a new protocol is added in the protocol table.

key type	name	datatype	linked to table
primary key	protocolID	int	-
foreign key	siteID	int	site

Table 5: protocol table keys

column	datatype	description	outdated by birdscan v1.6
protocolID	int	protocol ID	no
siteID	int	site ID	no
startTime	datetime	start time and date of the protocol	no
stop_time	datetime	stop time and date of the protocol	no
pulseType	varchar	pulsetype: short, medium, long (“S”,”M”,”L”)	no
rotate	bit	radar in rotation mode (1/0)	no
stc	float	STC altitude in m	no
threshold	float	stc threshold in dBm	no
blockTime	float	time radar is blind after protocol change in secs	no

Table 6: important columns in protocol table

2.2.4 visibility

The visibility table store the times when the radar was blind due to weather conditions (rain, snow, heavy fog), clutter (e.g.: radome covered with snow or leaves, too many objects like trees, houses or vehicles nearby) or protocol changes. Blindtimes in the visibility table are linked to one protocol and are thus split if expanding over two or more protocols.

key type	name	datatype	linked to table
primary key	visibilityLogID	int	-
foreign key	protocolID	int	protocol

Table 7: visibility table keys

column	datatype	description	outdated by birdscan v1.6
visibilityLogID	int	visibility ID	no
protocolID	int	link to protocol table	no
blind_from	datetime	start time from when the radar was blind	no
blind_to	datetime	stop time to when the radar was blind	no

Table 8: important columns in visibility table

2.2.5 site

The site table contains information about the site where the radar is placed. One row per site.

key type	name	datatype	linked to table
primary key	siteID	int	-
foreign key	radarID	int	radar

Table 9: site table keys

column	datatype	description	outdated by birdscan v1.6
siteID	int	site ID	no
siteCode	varchar	3-letters abbreviation of the sitename	no
radarID	smallint	radar ID	no
siteName	varchar	name of the site	no
siteDesc	varchar	description of the site	no
projectStart	date	startdate of the project/campaign	no
projectEnd	date	enddate of the project/campaign	no
longitude	float	longitude of the site location in decimal degrees	no
latitude	float	latitude of the site location in decimal degrees	no
altitude	smallint	altitude a.s.l. of the site location in m	no
timeShift	varchar	time shift/zone at the site location	no
radarOrientation	float	orientation of the radar to north on site	no

Table 10: important columns in site table

2.2.6 radar

The radar table contains information about the radar hardware. For general analysis these values are not of interest.

key type	name	datatype	linked to table
primary key	radarID	int	-

Table 11: radar table keys

column	datatype	description	outdated by birdscan v1.6
radarID	smallint	radar ID	no
type	varchar	radar Type	no
serialNo	smallint	serial number	no

Table 12: important columns in radar table

2.2.7 weather and weather_property

The weather table contains information about the environment. The temperature/humidity sensors are placed inside the radar (one inside the radome and one behind the ventilation inlet) and do not represent the outside air temperature and humidity. The weather table is linked to the time_bins table and the weather_property table. One row per time bin and weather property is created.

The weather_property table describes all weather properties.

key type	name	datatype	linked to table
primary/foreign key	time_bin	int	time_bins
primary/foreign key	weather_property	int	weather_property

Table 13: weather table keys

key type	name	datatype	linked to table
primary key	id	int	-

Table 14: weather_property table keys

property	id	unit
Temperature Radome	1	degree C
Temperature Fresh Air	2	degree C
Relative Humidity Radome	3	%
Relative Humidity Fresh Air	4	%
Blind Time Percent	5	%

Table 15: weather properties

2.2.8 echo_rffeatures_map and rffeatures

The rffeatures table lists all features that are computed by the algorithms. A selection of them is used by the random forest models for classification and wing beat frequency estimation. The wing beat frequency and its credibility are part of the rffeatures and thus saved in the echo_rffeatures_map table. The echo_rffeatures_map contains one row per echo and rffeature.

The features are roughly described in chapter 2.3.

key type	name	datatype	linked to table
primary/foreign key	echo	int	collection
primary/foreign key	feature	int	rffeatures

Table 16: echo_rffeatures_map table keys

key type	Name	datatype	linked to table
primary key	Id	int	-

Table 17: rffeatures table keys

2.2.9 echo_validation and echo_validation_type

The echo validation algorithm of the BirdScan software validates each echo, if it is a valid bio scatterer or non-bio scatterer like rain, snow or ground clutter. The two types are listed in the echo_validation_type table and in the echo_validation table one row per echo is listed with its validation.

The echo validation algorithm is only working with raw radar data; therefore, the echo validation is not available for reclassified databases created by BirdScan versions older than v1.6.

key type	name	datatype	linked to table
primary/foreign key	echo_id	int	collection
foreign key	type	int	echo_validation_type

Table 18: echo_validation table keys

key type	name	datatype	linked to table
primary key	id	int	-

Table 19: echo_validation_type table keys

property	id
bio scatterer	1
non-bio scatterer	2

Table 20: echo validation types

2.2.10 rf_classification and rfclasses

the rf_classification table contains the classification based on the random forest classifier released with BirdScan version v1.6. The available classes are listed in the rfclasses table. The rfclasses table lists more classes than used by the classifier. Refer to the column 'isUsedForClassification' to get the classes used by the classifier.

key type	name	datatype	linked to table
primary/foreign key	echo	int	collection
foreign key	class	int	rfclasses

Table 21: rf_classification table keys

key type	name	datatype	linked to table
primary key	id	int	-

Table 22: rfclasses table keys

column	datatype	description	outdated by birdscan v1.6
echo	int	echo ID, link to collection table	no
class	int	class ID, link to rfclasses table	no
mtr_factor	real	MTR factor based on the random forest classifier	no
class_probability	real	classification probability	no

Table 23: important columns in rf_classification table

2.2.11 rf_class_probability

The rf_class_probability table is like the rf_classification table linked to the collection and rfclasses table. It lists the probabilities as well as the MTR factors for all classes. The MTR factor is dependent on the size of the object (RCS). For small RCS, the RCS is replaced by a typical size for the class. Therefore, the MTR factor for one echo maybe different according to the classification.

This table exists for the case if one wants to change classifications manually, the probabilities and MTR factors do not have to be recalculated.

The table contains one row per echo and class.

key type	**name**	**datatype**	**linked to table**
primary/foreign key	echo	int	collection
primary/foreign key	class	int	rfclasses

Table 24: rf_class_probability table keys

2.3 RF Features

The radar computes a set of features which are used for the random forest models of the wing beat frequency estimator and the classifier. Both the wing beat frequency and the classifier use a subset of the features.

2.3.1 Statistical Features

The following statistical values are computed both for the raw and bandpass signal:

- Sample standard deviation
- Skewness
- Kurtosis

Additionally, the duration of the signal is saved.

#	Feature	DB-Name	ID
1	duration of the signature	duration_sec	0
2	standard deviation of the raw signal	std_sigRaw2	1
3	skewness of the raw signal	skw_sigRaw2	2
4	kurtosis of the raw signal	krt_sigRaw2	3
5	standard deviation of the band-passed signal	std_sigBP	4
6	skewness of the band-passed signal	skw_sigBP	5
	kurtosis of the band-passed signal	krt_sigBP	6

Table 25: basic features

2.3.2 Overall Shape Features

The overall shape features represent the changes respectively the consistency of the signal over time.

The signal is divided into time chunks. Within each chunk, the mean of the raw signal and the standard deviation of the bandpass signal is computed. Afterwards, the mean and standard deviation differences from one chunk to another are computed. The minimum, maximum and mean value of the differences are used as features.

#	Feature	DB-Name	ID
1	min difference of mean	shapeFeatures_1	7
2	mean difference of mean	shapeFeatures_2	8
3	max difference of mean	shapeFeatures_3	9
4	min difference of std	shapeFeatures_4	10
5	mean difference of std	shapeFeatures_5	11
6	max difference of std	shapeFeatures_6	12

Table 26: overall shape features

2.3.3 STFT Features

The STFT (Short Time Fourier Transform) features are based on the frequency spectrum of the high-pass power signal (in watt), calculated via the STFT. The different features are:

- The simple features are the entropy of the spectrum, the frequency at the maximum of the spectrum as well as the moments.
- The frequency at the maximum peak of the harmonic product spectrum and its entropy is computed for different numbers of down sampling factors.
- The peak detection computes different values of the peaks in the spectrum, such as the frequency of the highest peak, the relative height of it, the uniqueness of the highest peak and the inverted variance of the frequency differences of all detected peaks.
- For the broad bin spectrum, the STFT spectrum is reduced (averaged) to fewer, broader bins (default 30 bins).

#	Feature	DB-Name	ID
1	entropy of the spectrum	spec_entropy	13
2	frequency of the maximum value of the spectrum	spec_mod_freq	14
3	mean of the spectrum	spec_mea_freq	15
4	standard deviation of the spectrum	spec_std_freq	16
5	skewness of the spectrum	spec_skw_freq	17
6	kurtosis of the spectrum	spec_krt_freq	18
7	frequency of the highest peak in the spectrum	StftFreqOfHighestPeak	19
8	relative height of the highest peak in the spectrum	StftRelHeightOfHighestPeak	20
	uniqueness of the highest peak in the spectrum	StftUniquenessOfHighestPeak	21

	inverted variance of frequency differences between the peaks	StftInvVarOfPeakDiffs	22
	fundamental frequency of harmonic product spectrum	HPSfundFreHz1 - 10	23 - 32
	entropy of harmonic product spectrum	HPSentropy1 - 10	33 - 42
	Broad Bins	BroadSpec_1 - 30	43 - 72

Table 27: STFT features

2.3.4 Auto Correlation Features

The autocorrelation features are based on the high-pass power (in watt) signal. The following features are computed:

- A sine-wave regression (least square) is processed on the autocorrelation. The sine wave frequency of the best fit and its regression error are used as features.
- A peak detection detects the highest peak, first peak and first peak above zero. The frequency and height of the peaks are used as features.

Structure in the Features class

#	Feature	DB-Name	ID
1	sine wave regression frequency	ACSineRegrFreq	73
2	sine wave regression error	ACSineRegrError	74
3	frequency of first peak	ACFirstPeakFreq	75
4	height of first peak	ACFirstPeakHeight	76
5	frequency of highest peak	ACMaxPeakFreq	77
6	height of highest peak	ACMaxPeakHeight	78
7	frequency of first peak above zero	ACFirstPeakAboveZeroFreq	79
8	height of first peak above zero	ACFirstPeakAboveZeroHeight	80

Table 28: auto correlation features

2.3.5 Wing Flapping Packages

To detect the wing flapping packages (wfp), the bandpass signal is used as base. The band-pass signal is converted to absolute values. To compute the wfp, several filters are applied:

- Moving mean of the absolute bandpass signal with a narrow window (0.15s)
- Moving maximum of the moving mean with a wide window (2.25s) for the upper envelope
- Moving minimum of the moving mean with a wide window (2.25s) for the lower envelope
- Moving ascending/descending test of the moving mean to indicate if the moving mean is varying or only ascending or descending

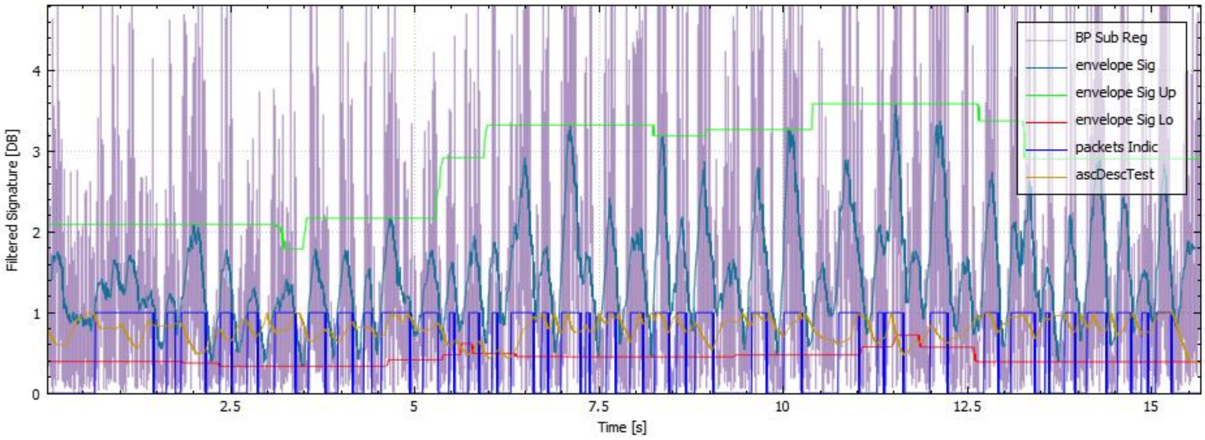

Figure 2: wing flapping packages

To detect the flapping and pause phases, different values are compared to thresholds:

- Ratio between lower and upper envelope
- Ratio between upper envelope and absolute maximum of upper envelope
- Moving mean (compared to a local threshold)
- Ascending/descending test

If those four conditions are true, a sample of the signal is considered as flapping, otherwise as pause. This creates a binary signal and based on this, the wfp features are computed. These are:

- Proportion of flapping duration per signal duration
- Number of flapping packets, mean duration and standard deviation
- Number of pause packets, mean duration and standard deviation

Additionally, the spectrum of moving mean of the absolute bandpass signal (0.15s window) is computed via the STFT, and the lowest 28 frequency bins are used as features (ampiSpecFeat).

Structure in the Features class:

#	Feature	DB-Name	ID
1	proportional feature phase	WFPf_proportionFeat_phase	113
2	number of packets phase	WFPf_numP_phase	114
3	mean duration phase packets	WFPf_meadur_phase	115
4	duration of packets phase, standard deviation	WFPf_cv_dur_std_phase	116
5	number of packets pause	WFPf_numP_pause	117
6	mean duration pause packets	WFPf_meadur_pause	118
7	duration of packets pause, standard deviation	WFPf_cv_dur_std_pause	119
8	ampli spec features	ampliSpecFeat1_1 - 28	81 - 108

Table 29: WFP features

2.3.6 RCS Features

For the RCS features the RCS values based on the raw signal and the lowpass signal are computed. The following three values are used as features:

- RCS value of the maximum of the lowpass signal
- RCS value of the maximum of the raw signal
- RCS value of the mean of the raw signal

#	Feature	DB-Name	ID
1	nearest RCS	RCS2_RCS_nearest_fea14	109
2	max lowpass RCS	RCS2_RCS_max_lowpassed	110
3	max raw RCS	RCS2_RCS_max_rawsignal	111
4	mean raw RCS	RCS2_RCS_mea_rawsignal	112

Table 30: RCS features

2.3.7 Low Frequency Features

The low frequency features are based on the spectrum computed via the STFT (mean over frequency bins) of the high-pass signal. The following values are used as features:

- Harmonic product spectrum (down sampling factors 1-4): frequency of the maximum peaks of the hps and its entropy
- Frequency bins 1-11 of the spectrum (low frequency bins)
- Frequency bins (width 2Hz) within a range (4-50Hz, medium frequency bins)

#	Feature	DB-Name	ID
1	Orientation/Phi	loHPSfundFreHz_1 - 4	159 - 162
2	mean RCS - A0	loHPSentropy_1 - 4	163 - 166
3	elongation -A2	spectrLowBins_1 - 11	122 - 132
4	cruciform element – A4	spectrMedBins_1 - 23	133 – 155

Table 31: low frequency features

2.3.8 Polarization / Shape Features

The polarization features are based on the raw and north signal and is only computed for echoes detected in modes with radar rotation.

The raw signal is split in 360° parts (indicated by the north signal). If the north signal is not available, the split can be approximated with the rotation frequency of the echo, given in the database. The mean over all splits is computed and normalized by its mean value. The parameters of the formula are fitted to the normalized mean split with a least square regression.

$$\|\sigma_{RCS}(\Omega t)\| = a_0 + a_2 \cdot \cos\big(2 \cdot (\Omega \mathrm{t} - \varphi)\big) + a_4 \cdot \cos\big(4 \cdot (\Omega \mathrm{t} - \delta)\big)$$

$$a_0 : \sim avg\ rcs\ (= 1, normalized)$$
$$a_2 : \sim elongation$$
$$a_4 : \sim cruciform\ element$$
$$\varphi : \sim orientation$$
$$\delta : \approx \varphi\ (insect\ symmetry\ axis)$$

Formula 1: polarization feature formula

The parameters of the best fit and its regression error are used as features.

#	Feature	DB-Name	ID
1	Orientation/Phi	shapeFeatPhi	169
2	mean RCS - A0	shapeFeatA0	170
3	elongation -A2	shapeFeatA2	171
4	cruciform element – A4	shapeFeatA4	172
5	regression error	shapeFeatRegrErr	173

Table 32: polarization/shape features

2.3.9 Wing Beat Frequency

The wing beat frequency is computed based on a subset of the features described in chapter 2.3 and saved itself as a feature. Along with the wing beat frequency a credibility value (0…1) is saved, quantifying the reliability of the wing beat frequency computation.

#	Feature	DB-Name	ID
1	wing beat frequency	WFF_predicted	167
2	credibility	WFF_credibility	168

Table 33: wing beat frequency and credibility