



AQUAculture **USE**r driven operational
Remote Sensing information services

Deliverable 4.2

HAB detection software

PML, WI, FFCUL, NIVA, GRAS

2014-12



AQUA-USERS is funded under the European Community's 7th Framework Program (Theme SPA.2013.1.1-06: Stimulating development of downstream services and service evolution, Grant Agreement N° 607325)

Task 4.2: Implementation of HAB detection method

Deliverable 4.2: HAB detection software

Lead beneficiary	PML
Contributors	WI, FFCUL, NIVA, GRAS
Due date	30/11/2014
Actual submission date	15/12/2014
Dissemination level	PU

Change record

Issue	Date	Change record	Authors
0.1	24/11/14	Document outline	PML
0.2	28/11/14	First draft	PML
1.0	10/12/14	Final Editing	PML, GRAS, WI, NIVA, VU/VUmc

Consortium

No	Name	Short Name
1	Water Insight BV	WI
2	Stichting VU-VUMC	VU/VUmc
3	Plymouth Marine Laboratory	PML
4	Fundação da Faculdade de Ciências da Universidade de Lisboa	FFCUL
5	Norsk institutt for vannforskning	NIVA
6	DHI-GRAS	GRAS
7	DHI	DHI
8	Sagremarisco-Viveiros de Marisco Lda	SGM

To be cited as

Kurekin, A., Miller, P., Poser, K., Eleveld, M., Dale, T. (2014) "HAB detection software", AQUA-USERS deliverable D4.2, EC FP7 grant agreement no: 607325, 17p.

Task objective (from DoW)

- Transforming HAB detection method into an operational tool
- Region specific training of HAB classifier for different harmful phytoplankton species
- Integration of developed tool into the satellite data processing chain at PML

Scope of this document

This document provides technical details of the HAB detection method developed within the AQUA-USERS project. The methodology for HAB detection has been developed in Task 3.3 and will be documented in Deliverable 3.3 “HAB detection methods report”. This document describes user requirements; the concept of HAB detection tool; the input, output and data storage formats; programming language; testing and validation process; software implementation of training and detection algorithms; integration into a satellite data processing chain at PML.

Abstract

The implementation of the HAB detection tool is based on the methodology being developed in Task 3.3. The constructed prototype of the HAB tool has been tested on MERIS data, correctly identifying 89% of *Phaeocystis globosa* HABs in the southern North Sea and 88% of *Karenia mikimotoi* blooms in the western English Channel. The tool can be easily extended to new HAB species, new regions and sensors without modifying its architecture or program code.

List of abbreviations

Abbreviation	Description
EO	Earth Observation
HAB	Harmful Algal Bloom
IOP	Inherent Optical Properties
LDA	Linear Discriminant Analysis
MERIS	MEdium Resolution Imaging Spectrometer
OLCI	Ocean Land Colour Instrument
Rrs	Remote Sensing Reflectance
SDA	Stepwise Discriminant Analysis

List of related documents

Short	Description	Date
DoW	Grant Agreement Annex I (Description of Work)	01/07/13
D2.1	Initial user requirements	31/01/14
D2.2	Technical design + set up satellite data infrastructure	28/02/14
D2.6	Data policy, data communication and access protocols	30/04/2014
D4.1	MERIS + S3 products archive	05/09/14

Table of contents

1	Introduction.....	6
2	Requirement analysis.....	7
3	HAB tool design	8
3.1	Feature selection.....	8
3.2	Classification method.....	9
4	Tool implementation	10
4.1	Training module	10
4.2	Module for classification.....	12
4.3	Integration into the satellite data processing chain at PML.....	13
4.4	Testing and validation	13
5	Conclusions.....	15
6	References.....	16

1 Introduction

Information about harmful algal blooms is considered as highly important and valuable for management in the aquaculture industry. In high concentrations harmful algae may cause respiratory irritation for humans, shellfish neurotoxic poisoning, fish mortality and can lead to significant economic losses by damaging fish farms (Stumpf *et al.*, 2003; Miller *et al.*, 2006). Consequently, significant efforts were put in AQUA-USERS towards timely detection of HAB events and providing this information along with other parameters of water quality to the users of aquaculture sites to assist them in taking management decisions. The HAB detection method, being developed in Task 3.3, uses spectral measurements of water leaving radiance to estimate harmfulness of algal blooms. It employs a fully automatic data-driven approach to identify key characteristics of water leaving radiances and derived quantities, and to classify pixels into “harmful”, “non-harmful” and “no bloom” categories. This report (Task 4.2) describes the software implementation of the HAB detection method. The HAB tool has been created that utilises a set of functions required for training the HAB method on different species and sensors., such as Sentinel-3 OLCI. Classification of satellite measurements into harmful and non-harmful categories and generation of HAB risk maps for aquaculture sites are carried out by processing historical data or data delivered in near-real time as part of satellite data processing chain at PML.

2 Requirement analysis

The questionnaire being developed and analysed in details in Deliverable 2.1 “Initial user requirements” clearly indicates the importance for HAB information for site selection and managing. More than 50% of users, who participated in questionnaire, applied analysis of harmful algal species for monitoring of their sites. The monitoring was carried out mostly by analysis of water samples under the microscope on a weekly basis. More than one harmful species were monitored on aquaculture sites and many users need more efficient and cheap monitoring technique to access information about harmful species on a regular basis. The questionnaire also demonstrated that users prefer to use maps for site selection tasks.

The access to historical data from the users’ sites and developed indicators of HAB stored in the EO data archive is regulated by the data policy specified in Deliverable 2.6. The document introduces three levels of sensitivity for the project data. As specified in Deliverable 2.2 “Satellite data infrastructure blue print”, the HAB risk maps generated by HAB tool are stored in the EO archive and will be accessed by users through the Web Portal. The authentication and restricted data access can be implemented when the users connect to the Web Portal.

The satellite data that will be processed to generate HAB risk maps are stored in the EO data archive. This includes MERIS Rrs products, relevant flags (land-water, clouds, coastline, etc.) and water quality products based on different Inherent Optical Property (IOP) algorithms optimised for the water type of each study region. The format of satellite EO data was specified in Deliverable 2.2 “Satellite data infrastructure blue print”. The EO data are stored in NetCDF version 4 file format and comply with the key remote sensing standards – the Climate and Forecast (CF) Conventions and the Unidata NetCDF discovery metadata conventions (CF Metadata, 2014). The OLCI Sentinel 3 data processed to Levels 3 and 4 will also follow the same file format. This unified format will simplify access to EO data archive and data import into the HAB detection application.

3 HAB tool design

The HAB classification tool implements HAB discrimination techniques described in Kurekin et al. (2014). It uses a fully automatic data-driven approach to identify key characteristics of remote sensing reflectance (R_{rs}) and derived quantities. The selected characteristics are then automatically classified into “harmful”, “not harmful” and “no bloom” categories. The tool can be trained on different HAB species and can be extended handle data from new ocean colour sensors. Applied to satellite images the HAB tool produces HAB risk maps being used for optimisation of site selection in case study 1 and generation of daily management decisions for aquaculture sites in case studies 2 and 3.

3.1 Feature selection

The main stages of the training procedure are illustrated in Figure 1. This includes selection and identification of training image scenes, sampling training data, feature selection and training the LDA classifier. Classification of training data is based on *in situ* sampling and visual interpretation (Miller et al., 2006) of ocean colour scenes and chlorophyll-a products.

The tool uses remote sensing reflectance $R_{rs}(\lambda)$ measurements at the available spectral bands as the main features to discriminate HAB species in water. The inherent optical properties (IOPs) of natural water bodies, such as the total absorption $a(\lambda)$ and backscattering $b_b(\lambda)$ coefficients are applied as additional features to improve the discrimination efficiency.

Additional features derived from the spectral ratio of two neighbouring wavelengths are used in the classifier to improve robustness to sensor calibration and atmospheric correction errors as well as to overcome limitations of linear classifiers (Duda et al., 2001). These features are calculated as the spectral ratios of reflectances $R_{Rrs}(i, j) = R_{rs}(\lambda_i) / R_{rs}(\lambda_j)$, absorption $R_a(i, j) = a(\lambda_i) / a(\lambda_j)$ and backscatter $R_b(i, j) = b_b(\lambda_i) / b_b(\lambda_j)$.

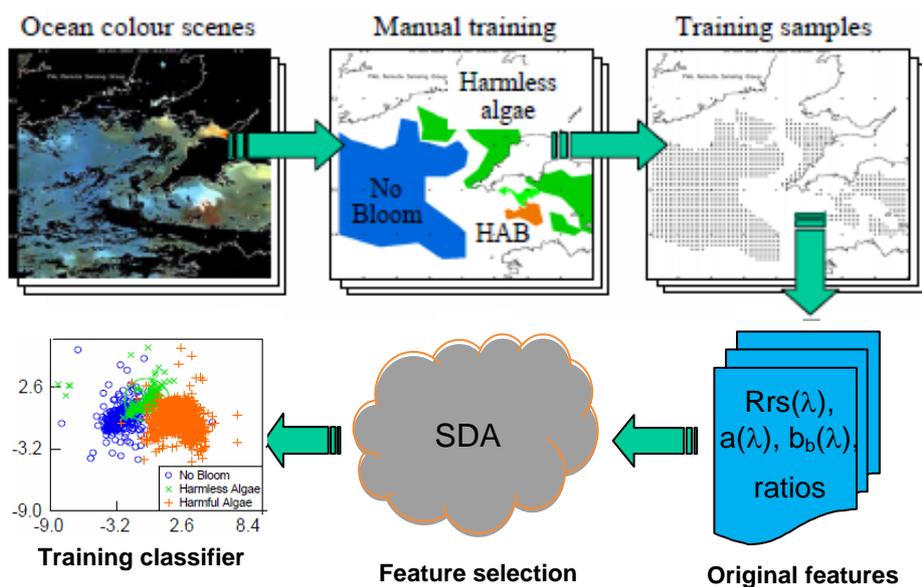


Figure 1: The main stages of training HAB tool.

The features are calculated on a pixel by pixel basis forming a three dimensional array with the first two dimensions corresponding to spatial coordinates and the third dimension corresponding to the feature vector.

A reduced number of most important features are selected at the final stage in Figure 1 to improve performance. This is achieved by applying a Stepwise Discriminant Analysis (SDA) algorithm

implemented in the statistical package 'klaR' (Weihs et al., 2005). It uses the probability of correct classification criterion to select the best combination of features iteratively by adding more significant or removing less significant features one by one.

3.2 Classification method

Classification of satellite measurements is carried out on a pixel by pixel basis. For each pixel in the satellite image the features selected at the training stage are calculated and processed by a Linear Discriminant Analysis (LDA) classifier (Duda *et al.*, 2001; Bandos *et al.*, 2009). The classifier estimates probabilities of "harmful bloom", "harmless bloom" and "no bloom" classes for each tested sample using Bayes' theorem (Duda *et al.*, 2001). Assuming that the distribution of features is normal, the probability for each class is calculated as

$$p(c_i / \mathbf{x}) = \frac{\exp(-0.5\|\mathbf{x} - \boldsymbol{\mu}_i\|^2)P(c_i)}{\sum_{j=1..3} \exp(-0.5\|\mathbf{x} - \boldsymbol{\mu}_j\|^2)P(c_j)}, i = 1..3, \quad (1)$$

where \mathbf{x} is a vector to be classified and $\boldsymbol{\mu}_i$ is the mean value of the i -th class in the discriminant coordinate system. Here $P(c_i)$ describes the *prior* probability for class c_i . The denominator provides the normalisation, such that the total probability to belong to one of the three classes always sums to 1.0.

To reduce the false alarm rate for input data that do not match any of the trained categories we introduced an additional "unknown" class. The probability for samples of this class is evaluated separately by calculating the distance d_{\min} between the input sample and the mean of the nearest class, and finding the probability that for the samples of this class the distance is less than d_{\min} . This probability becomes higher when the distances from mean value $\boldsymbol{\mu}_i$ is increased, so it can be considered as a measure of dissimilarity with the other three classes. Assuming that the distribution of features is Gaussian, the probability of "unknown" class is calculated as $p(c_u / x) = \chi_{cdf}^2(d_{\min}^2, n_d)$, where $\chi_{cdf}^2(\dots)$ is a cumulative density function of chi-squared distribution and n_d is the dimension of the discriminant vector ($n_d = 2$ for three classes). The "unknown" class is applied to identify areas where the results of LDA classification are unreliable.

4 Tool implementation

The HAB tool consists of two main functional modules. The first module has been developed for training the HAB classifier to discriminate new HAB species or to extend it to new sensors. This module implements functions for reading training data stored in a file, selecting features for HAB classification, training LDA classifier and storing training results in the configuration file.

The second module is applied for processing satellite data and HAB classification. It implements reading of satellite data from the EO data archive, classification of satellite images on pixel by pixel basis, generating HAB risk maps and saving them in NetCDF4 or PNG files.

The modules have been implemented in the Python 2.7 programming language using object oriented programming. The standard Python extension libraries have been applied to support operations with NetCDF4 files, multi-dimensional array and matrixes, interaction with the statistical package 'klaR'. The block diagram of HAB tool is illustrated in Figure 2.

The design of HAB tool presented in Figure 2 is not final and will be extended by including hyperspectral classification algorithm when hyperspectral measurements will become available for harmful species provided by the laboratory experiment in Task 3.3 and WISP-3 sensor measurements being carried out by the user community.

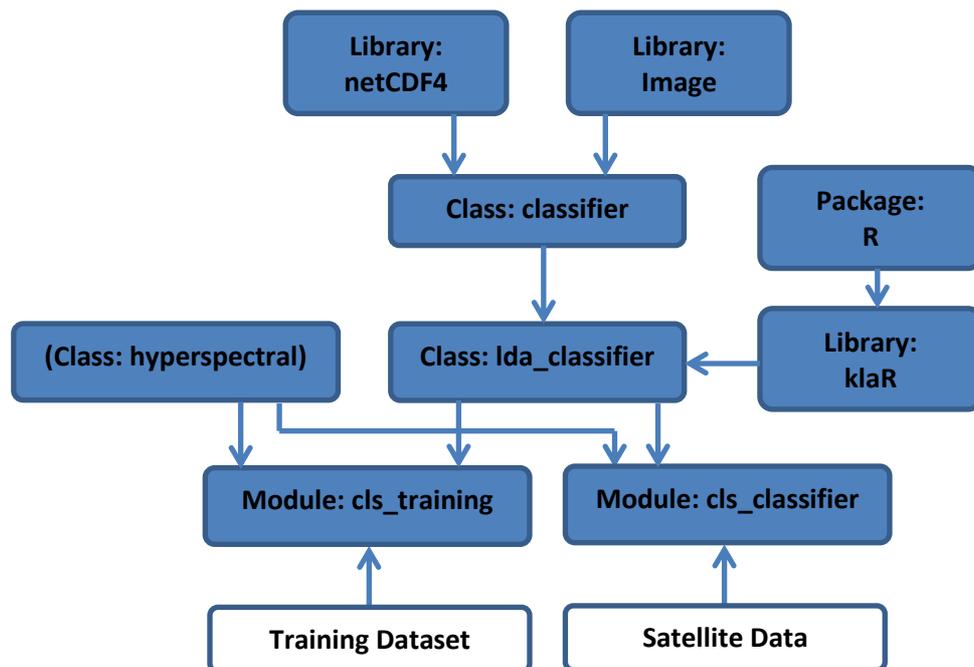


Figure 2: Block diagram of HAB tool

4.1 Training module

The training module is implemented as a Python script "cls_training". It uses a command line interface to specify input parameters:

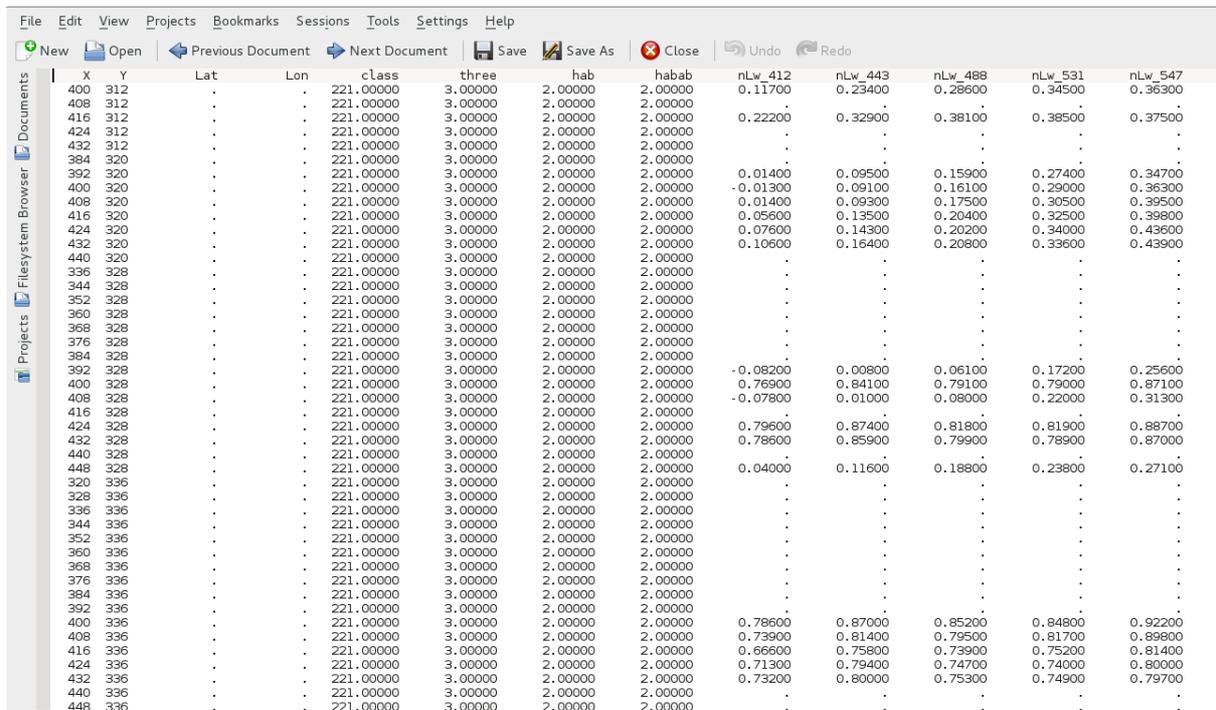
- the name for HAB species to discriminate (i.e. "karenia", "phaeocystis")
- the name of sensor (i.e. "meris")
- the name of the training data file
- the names of features applied for HAB classification (i.e. "Rrs_413", "a_413_pml", "bb_510_pml")

Data for training the HAB classifier are stored in a text file as shown in Figure 3. Each row in the table represents one sampling point in the satellite image. The information for sampling points is stored in the columns:

- X, Y: coordinates of sampled data in the image
- Lat, Lon: latitude and longitude coordinates
- class: an integer code of algal species
- hab: the flag indicating harmfulness
- Rrs and IOP measurements for different bands

A graphical representation of the training module is given in Figure 4. The sequence of function calls is indicated in Figure 4 by arrows. The main stages of training algorithm are implemented in function `train_classifier(...)` of class "classifier". It calls function "load_training_data(...)" to load training data from file. This function returns measurements of $Rrs(\lambda)$, $a(\lambda)$ and $b_b(\lambda)$ for the selected HAB species. It also returns additional features calculated as spectral ratios of reflectances, absorption and backscatter (see Section 3.1 for details). At the next stage the function "feature_selection(...)" is called, which implements the SDA feature selection algorithm described in Section 3.1. The selected features are returned by this function and then passed to the function "training_classifier(...)" to train LDA classifier. The accuracy of training is evaluated by calling function "cval_confusion_matrix(...)", which returns a confusion matrix of HAB classifier estimated by cross-validation method on training dataset.

At the final stage the configuration of the trained HAB classifier is stored in a text file by calling function "save_configuration(...)". The format of configuration file is shown in Figure 5. The file contains: a list of class names; prior probabilities for classes; a list of automatically selected classification features; the linear discriminant vector coefficients and mean vectors coefficients of LDA classifier; the confusion matrix and total probability of correct classification estimated for the training dataset.



X	Y	Lat	Lon	class	three	hab	habab	nLw_412	nLw_443	nLw_488	nLw_531	nLw_547
400	312	.	.	221.00000	3.00000	2.00000	2.00000	0.11700	0.23400	0.28600	0.34500	0.36300
408	312	.	.	221.00000	3.00000	2.00000	2.00000
416	312	.	.	221.00000	3.00000	2.00000	2.00000	0.22200	0.32900	0.38100	0.38500	0.37500
424	312	.	.	221.00000	3.00000	2.00000	2.00000
432	312	.	.	221.00000	3.00000	2.00000	2.00000
384	320	.	.	221.00000	3.00000	2.00000	2.00000
392	320	.	.	221.00000	3.00000	2.00000	2.00000
400	320	.	.	221.00000	3.00000	2.00000	2.00000	-0.01400	0.09500	0.15900	0.27400	0.34700
408	320	.	.	221.00000	3.00000	2.00000	2.00000	-0.01300	0.09100	0.16100	0.29000	0.36300
416	320	.	.	221.00000	3.00000	2.00000	2.00000	0.01400	0.09300	0.17500	0.30500	0.39500
424	320	.	.	221.00000	3.00000	2.00000	2.00000	0.05600	0.13500	0.20400	0.32500	0.39800
432	320	.	.	221.00000	3.00000	2.00000	2.00000	0.07600	0.14300	0.20200	0.34000	0.43600
440	320	.	.	221.00000	3.00000	2.00000	2.00000	0.10600	0.16400	0.20800	0.33600	0.43900
336	328	.	.	221.00000	3.00000	2.00000	2.00000
344	328	.	.	221.00000	3.00000	2.00000	2.00000
352	328	.	.	221.00000	3.00000	2.00000	2.00000
360	328	.	.	221.00000	3.00000	2.00000	2.00000
368	328	.	.	221.00000	3.00000	2.00000	2.00000
376	328	.	.	221.00000	3.00000	2.00000	2.00000
384	328	.	.	221.00000	3.00000	2.00000	2.00000
392	328	.	.	221.00000	3.00000	2.00000	2.00000	-0.08200	0.00800	0.06100	0.17200	0.25600
400	328	.	.	221.00000	3.00000	2.00000	2.00000	0.76900	0.84100	0.79100	0.79000	0.87100
408	328	.	.	221.00000	3.00000	2.00000	2.00000	-0.07800	0.01000	0.08900	0.22000	0.31300
416	328	.	.	221.00000	3.00000	2.00000	2.00000
424	328	.	.	221.00000	3.00000	2.00000	2.00000	0.79600	0.87400	0.81800	0.81900	0.88700
432	328	.	.	221.00000	3.00000	2.00000	2.00000	0.78600	0.85900	0.79900	0.78900	0.87000
440	328	.	.	221.00000	3.00000	2.00000	2.00000
448	328	.	.	221.00000	3.00000	2.00000	2.00000	0.04000	0.11600	0.18800	0.23800	0.27100
320	336	.	.	221.00000	3.00000	2.00000	2.00000
328	336	.	.	221.00000	3.00000	2.00000	2.00000
336	336	.	.	221.00000	3.00000	2.00000	2.00000
344	336	.	.	221.00000	3.00000	2.00000	2.00000
352	336	.	.	221.00000	3.00000	2.00000	2.00000
360	336	.	.	221.00000	3.00000	2.00000	2.00000
368	336	.	.	221.00000	3.00000	2.00000	2.00000
376	336	.	.	221.00000	3.00000	2.00000	2.00000
384	336	.	.	221.00000	3.00000	2.00000	2.00000
392	336	.	.	221.00000	3.00000	2.00000	2.00000
400	336	.	.	221.00000	3.00000	2.00000	2.00000	0.78600	0.87000	0.85200	0.84800	0.92200
408	336	.	.	221.00000	3.00000	2.00000	2.00000	0.79900	0.81400	0.79500	0.81700	0.89800
416	336	.	.	221.00000	3.00000	2.00000	2.00000	0.66600	0.75800	0.73900	0.75200	0.81400
424	336	.	.	221.00000	3.00000	2.00000	2.00000	0.71300	0.79400	0.74700	0.74000	0.80000
432	336	.	.	221.00000	3.00000	2.00000	2.00000	0.73200	0.80000	0.75300	0.74900	0.79700
440	336	.	.	221.00000	3.00000	2.00000	2.00000
448	336	.	.	221.00000	3.00000	2.00000	2.00000

Figure 3: The training data format

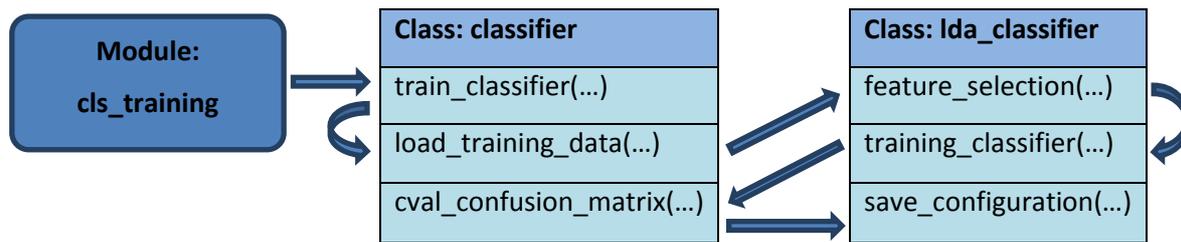


Figure 4: Graphical representation of HAB training module

```

*modis_lda_karenia_nc_Rrs.cfg x
[classifier]
class_names = ['nobloom', 'harmless', 'harmful']

feature_names = ['Rrs_488', 'Rrs_667', 'Rrs_443:Rrs_488', 'Rrs_488:Rrs_531', 'Rrs_531:Rrs_547']

prior_prob = array([ 0.45548166,  0.37824501,  0.16627333])

group_means = array([[ 4.02604872e-03,  3.25608108e-04,  9.37534206e-01,  1.66394919e+00,  1.26467369e+00],
 [ 3.54817242e-03,  2.91281094e-04,  7.03331204e-01,  1.17942510e+00,  1.13412264e+00],
 [ 1.74305186e-03,  6.90695880e-04,  5.39844585e-01,  7.31593014e-01,  9.06876308e-01]])

lin_discriminants = array([[ -6.67122730e+01,  -4.19181506e+02],
 [ -2.93848432e+02,  8.61562182e+02],
 [  3.16387540e-02,  6.95313665e-02],
 [ -3.38458499e+00,  4.41393035e+00],
 [ -2.79169803e+00,  -9.09084393e+00]])

confusion_matrix = array([[ 92.52840678,  6.95147047,  0.52012274],
 [ 23.3309304,  76.05541374,  0.61365586],
 [ 0.,  16.86884584,  83.13115416]])

p_total = 84.76746753515587

train_file_name = '/users/rsg/anku/projects/AquaMar/Classifier_data/training/tables/modis_data_all_nc_extra.txt'

```

Figure 5: Configuration file automatically generated by training module

4.2 Module for classification

The module for HAB classification has been implemented as a command line tool. It is written as a Python script that accepts several arguments:

- the name on classifier configuration file (see Figure 5)
- the name of satellite data file (in NetCDF4 format)
- additional options for the output HAB risk map

Graphical representation of the module is shown in Figure 6. Main stages of image classification are implemented in the function “classify_image(...)” defined within class “classifier”. It calls the function “load_configuration” to load the configuration of HAB classifier and the function “read_rsg_image(...)” to load data from the EO data archive in NetCDF format. These data are stored in a numpy array and then processed pixel by pixel by the function “classify(...)”. The probability of “no bloom”, “harmless bloom” and “harmful bloom” is estimated for image pixels and stored in a numpy array. This function also estimates probability of “unknown” class as described in Section 3.2. The arrays are then converted into HAB risk maps by the function “save_classified_image(...)”. HAB risk maps, generated for each class, are presented as a palette-based 8-bit images in the Portable Network Graphics (PNG) or NetCDF format with the estimated probabilities 0..1 mapped to 0..255 pixel values.



Figure 6: Graphical representation of HAB classification module

To simplify visual analysis of the classification results a colour composite HAB risk map is generated that represents probabilities of “harmful bloom”, “harmless bloom” and “no bloom” classes using a combination of red, green and blue primary colour components respectively, with the “unknown” class shown in grey. The values of red, green and blue components of a colour image are calculated as

$$\begin{aligned}
 R &= (p_{hab} \cdot (1 - p_{un}) + 0.5 \cdot p_{un}) \cdot 255 \\
 G &= (p_{hl} \cdot (1 - p_{un}) + 0.5 \cdot p_{un}) \cdot 255 , \\
 B &= (p_{nob} \cdot (1 - p_{un}) + 0.5 \cdot p_{un}) \cdot 255
 \end{aligned}
 \tag{2}$$

where p_{hab} , p_{hl} , p_{nob} and p_{un} are the risks of the respective classes. The saturation of colours in the composite map depends on the probability of the “unknown” class and for $p_{un} = 1$ the colour will be grey with $R = G = B = 128$. For high risk of “harmful bloom” and low probability of “unknown” class the colour will be red.

The HAB risk maps generated by the classification script are stored in the EO data archive in NetCDF format and will be accessed through the AQUA-USERS web portal. This imposes additional requirements on the data that will be displayed by the Web Portal. To fulfil these requirements, colour composite HAB risk maps, consisting of R, G and B components, will be converted into a one-component image with a palette. The palette will be optimized to display colour HAB risk maps with minimal distortions of colour in the original image.

4.3 Integration into the satellite data processing chain at PML

The developed HAB tool is configured for different operation areas, HAB species or sensors by simply changing its configuration file. This makes it simple to use for batch processing of historical MERIS data stored in the EO data archive and for integration into a satellite data processing chain developed by PML for near real time generation of HAB risk maps. The processing chain has been built around the SeaDAS data analysis system, proprietary modules in Python, C and C++ code, UNIX and Linux shell scripting languages. It is run on the Linux-based Sun Grid Engine that consists of 3 redundant master nodes and 72 computing nodes connected via Gigabit Ethernet to one another and approaching one petabyte of network attached storage.

The HAB tool was integrated into the PML processing chain and is called from a wrapper script “65_hab_classify.py” written in Python. The wrapper script controls the process priority of in the Grid Engine queue, automatically selects the HAB tool configuration file depending on sensor type and area name, supports logging by saving HAB tool messages to a file.

4.4 Testing and validation

The HAB tool has been tested on historical events of *Karenia mikimotoi* HAB being observed in the Western English Channel area. The *Karenia mikimotoi* training data were selected by visually analysing and labelling *Karenia* bloom events that occurred in years 2002-2004 in the MERIS images

(see **Fout! Verwijzingsbron niet gevonden.**). The labelled bloom regions in MERIS scenes were subsampled to produce a training dataset. The training data have been stored in a separate file, as shown in Figure 3. These data have been used for training HAB classifier to discriminate *Karenia* bloom. The classifier performance was estimated on training data by using cross-validation method with the total probability of correct classification $P_c > 87\%$.

Validation of the HAB tool was based on an analysis of a *Karenia* bloom in the Western English Channel in June-July 2010. The MERIS scene on 26 June 2010 was processed by the HAB tool configured to discriminate *Karenia* species. The *Karenia* HAB risk map generated by the HAB tool is shown in Figure 7. The map clearly demonstrates the presence of HAB in the Western English Channel to the east of Northern Brittany and Cornwall. In Figure 7 this region is shown in red indicating high risk of HAB while the other areas with low risk are shown in green and blue.

The HAB tool will be tested and validated on *Phaeocystis globosa* species when MERIS satellite data for the Southern North Sea will become available by W1. The scenes for validation of *Phaeocystis globosa* blooms will be selected in Task 3.3 by analysis of in-situ data obtained in 2003-2008 by measurement stations positioned along the Dutch coastal zone.

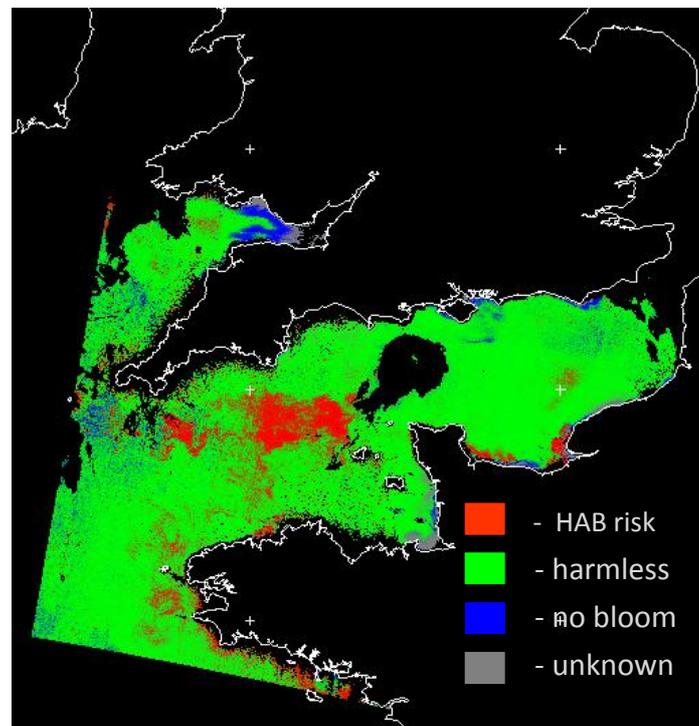


Figure 7: *Karenia* HAB risk map, 26 June 2010

5 Conclusions

This report presents the implementation of the HAB detection tool that will be used for processing historical MERIS data archive to generate HAB risk maps for the users' aquaculture sites. The tool has been integrated into the PML near real time processing chain, so that it can be used for processing Sentinel-3 OLCI data when they will become available. It has been successfully tested on *Karenia mikimotoi* bloom event in the Western English Channel in summer 2010.

The proposed implementation of HAB tool is not final and will be extended for the WISP-3 sensor when sufficient amount of hyperspectral measurements of HAB will be collected by the project partners. Alternatively the experimental measurements will be provided by the laboratory HAB measurement experiment being carried out in Tasks 3.3 and 3.5.

Table 1: The scenes used for training and independent testing of HAB tool

Region	Date	Sensor	Bloom types
Training scenes			
UK Southwest Approaches	19,22,28 July 2002, 16 August 2002	MERIS	<i>K. mikimotoi</i>
UK Southwest Approaches	20,25 June 2003, 1,7 July 2003, 11 August 2003	MERIS	<i>K. mikimotoi</i>
UK Southwest Approaches	10,22,23 July 2004, 30 August 2004, 2 September 2004	MERIS	<i>K. mikimotoi</i>
Validation scenes			
UK Southwest Approaches	22,26 June 2010	MERIS	<i>K mikimotoi</i>

6 References

- Bandos, T.V., Bruzzone, L. and Camps-Valls, G. (2009). Classification of hyperspectral images with regularized linear discriminant analysis, *IEEE Transactions on Geoscience and Remote Sensing*, 47(3), 862-873.
- CF Metadata. NetCDF Climate and Forecast Metadata Convention. [online] Available at <<http://cfconventions.org/>> [Accessed 27 November 2014]
- Duda, R.O., Hart, P.E., Stork, D.G. (2001). *Pattern classification*. (2nd ed.), Wiley, New York.
- Kurekin A., Miller P., Woerd J. (2014). Satellite discrimination of *Karenia mikimotoi* and *Phaeocystis* harmful algal blooms in European coastal waters: merged classification of ocean colour data. *Harmful Algae*, 31, 163-176.
- Miller, P.I., Shutler J. D., Moore G. F., Groom S. B. (2006). SeaWiFS discrimination of harmful algal bloom evolution. *Int. J. Remote Sens.*, 27(11), 2287-2301.
- Stumpf, R.P., Culver, M.E., Tester, P.A., Tomlinson, M.C., Kirkpatrick, G.J., Pederson, B.A., Truby, E., Ransibrahmanakul, V., Soracco, M. (2003). Monitoring *Karenia brevis* blooms in the Gulf of Mexico using satellite ocean color imagery and other data. *Harmful Algae*, 2(2), 147–160.
- Weih, C., Ligges, U., Luebke, K., Raabe, N. (2005). *klaR Analyzing German Business Cycles*. In Baier, D., Decker, R., Schmidt-Thieme, L. (Eds.). *Data Analysis and Decision Support*, (pp. 335-343). Springer-Verlag, Berlin.