Artificial Neural Networks

Cédric Jamet
Laboratory of Oceanology and Geosciences
Summer School ML4Oceans
cedric.jamet@univ-littoral.fr

Content

- **Two types of learning problem**
  - Supervised: Multi-Layer Perceptrons
  - Unsupervised: Self-Organizing Maps

Content

- **Two types of learning problem**
  - Supervised: Multi-Layer Perceptron
    - Provided with predictor data x_n and the response data y_n ($y=f(x)$)
    - Given predictor data as input, the model produces outputs y'_n
    - Model learning is "supervised" in the sense that the model output y'_n is guided towards the given response data y_n, usually by minimizing an *objective function* (cost function or error function)
    - Regression and classification problems involve supervised learning

Content

- **Two types of learning problem**
  - Unsupervised: Self-Organizing Maps
    - Only input data provided
    - Model discovers the natural patterns or structure in the input data
    - Clustering and principal component analysis involve unsupervised learning

Multi-Layer Perceptrons

What is a neuron?

- Definition: a neuron is a linear function, parameterised, with bounded values

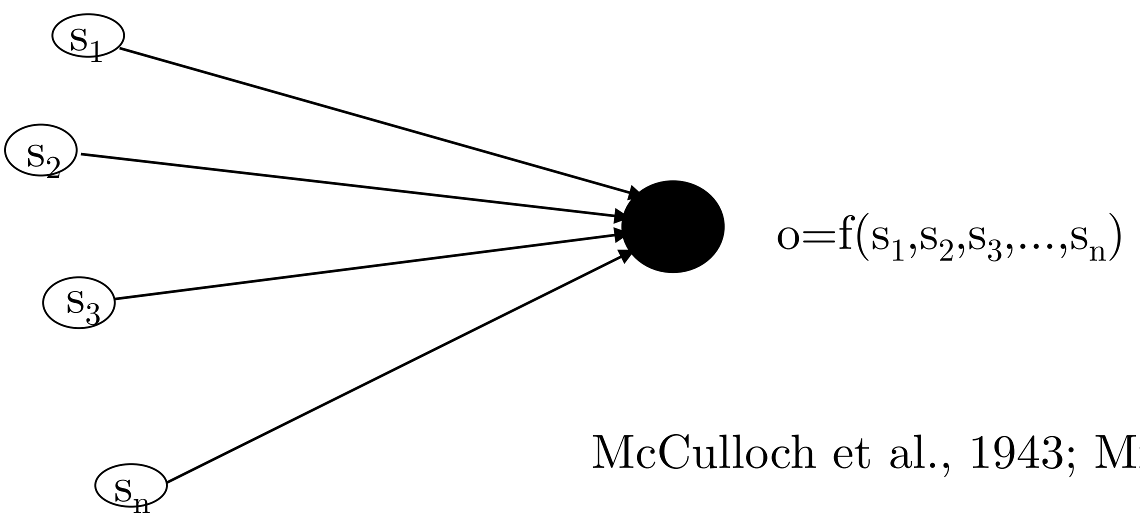
- A neuron is defined by three characteristics:
  - its state
  - its connexions with the others neurons
  - its activation (or transfer) function

What is a neural network?

- A neural network consists of a system of simple interconnected neurons, or nodes, which is a model representing a nonlinear mapping between an input vector and an output vector
- **Definitions:**
  - **O**: the ensemble of the possible states of the neurons
  - $\mathbf{o_i}$: the state of the i-th neuron o_i
  - a neighbourhood $\mathbf{s_1,\ldots, s_n}$
  - a transfer function associated to the neuron $\mathbf{f_i}$
  - $\mathbf{w_{ij}}$, the weight of the connexion from the j-th neuron toward the i-th neuron; w_{in} in W the ensemble of the weights

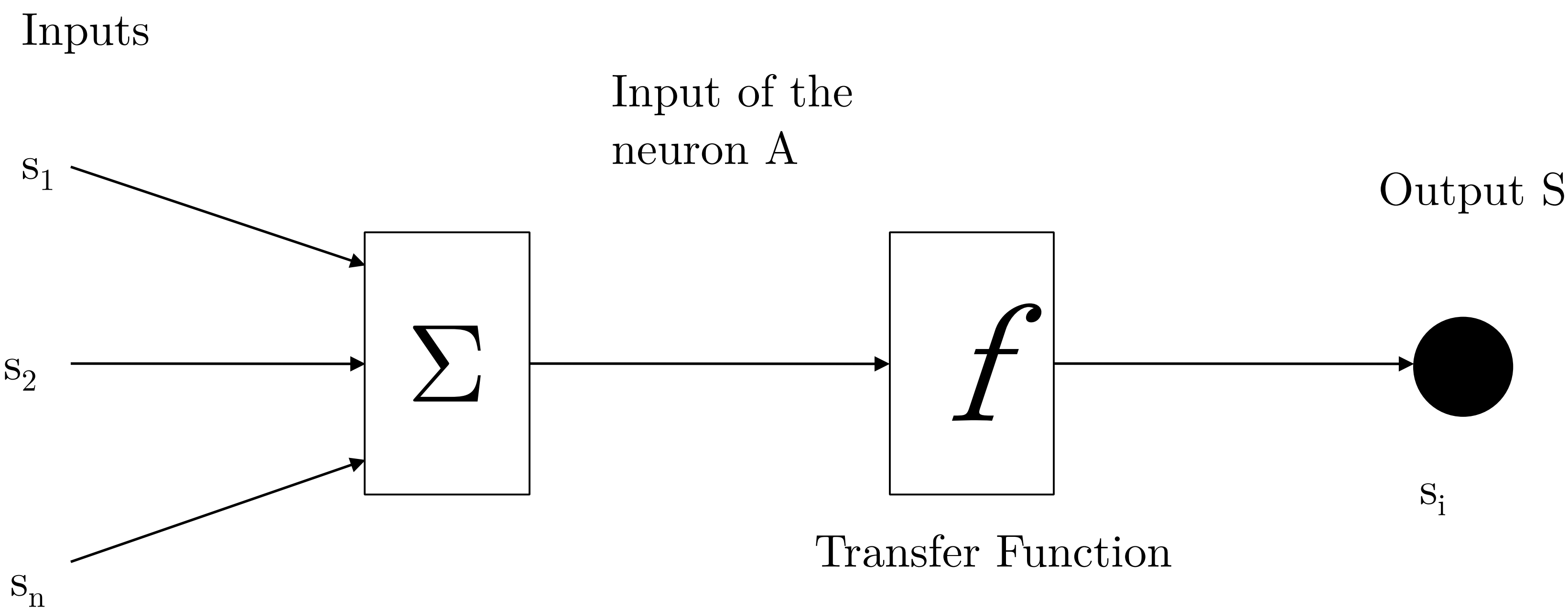
What is a neural network?

- Examples:
  - S=[0,1] ou [-1,1] binary neuron
    - 1: active
    - 0/-1: unactive
  - S=[0,1,2,....,k] discrete neuron
  - S=[a,b] continuous neuron



McCulloch et al., 1943; Minsky et al., 1969

Activation function f



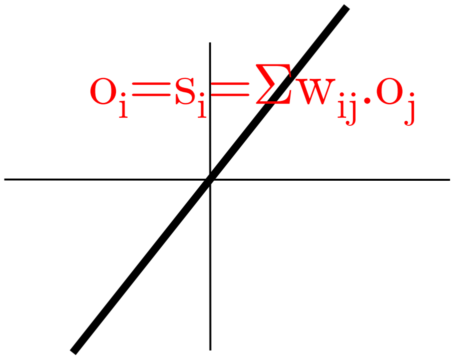
$o_i = f(s_i)$

$s_i = \Sigma w_{ij} o_j + w_{i0}$

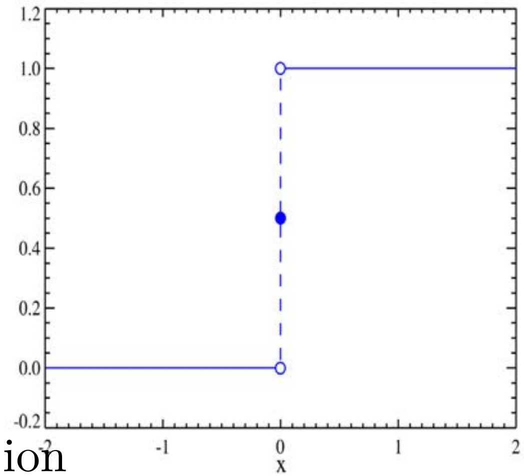
w_{ij} is a real, represents the weight of the connection between o_i and o_j

w_{i0}: bias/offset parameter

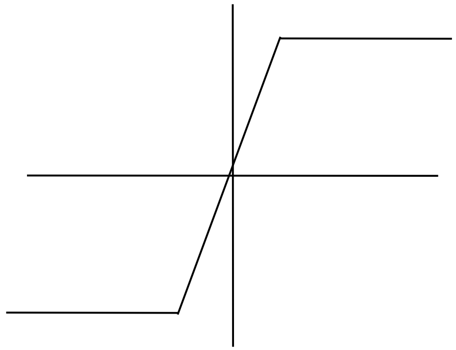
Differents types of activation function



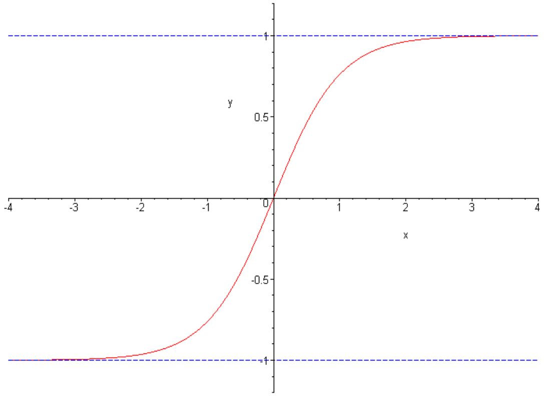
Linear function (or identity function



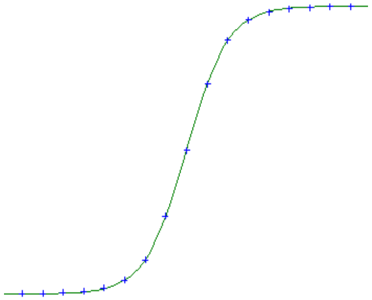
Heaviside Function



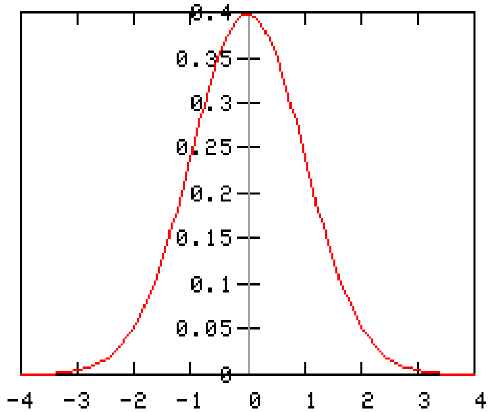
Quasi-Linear function



Hyperbolic tangent

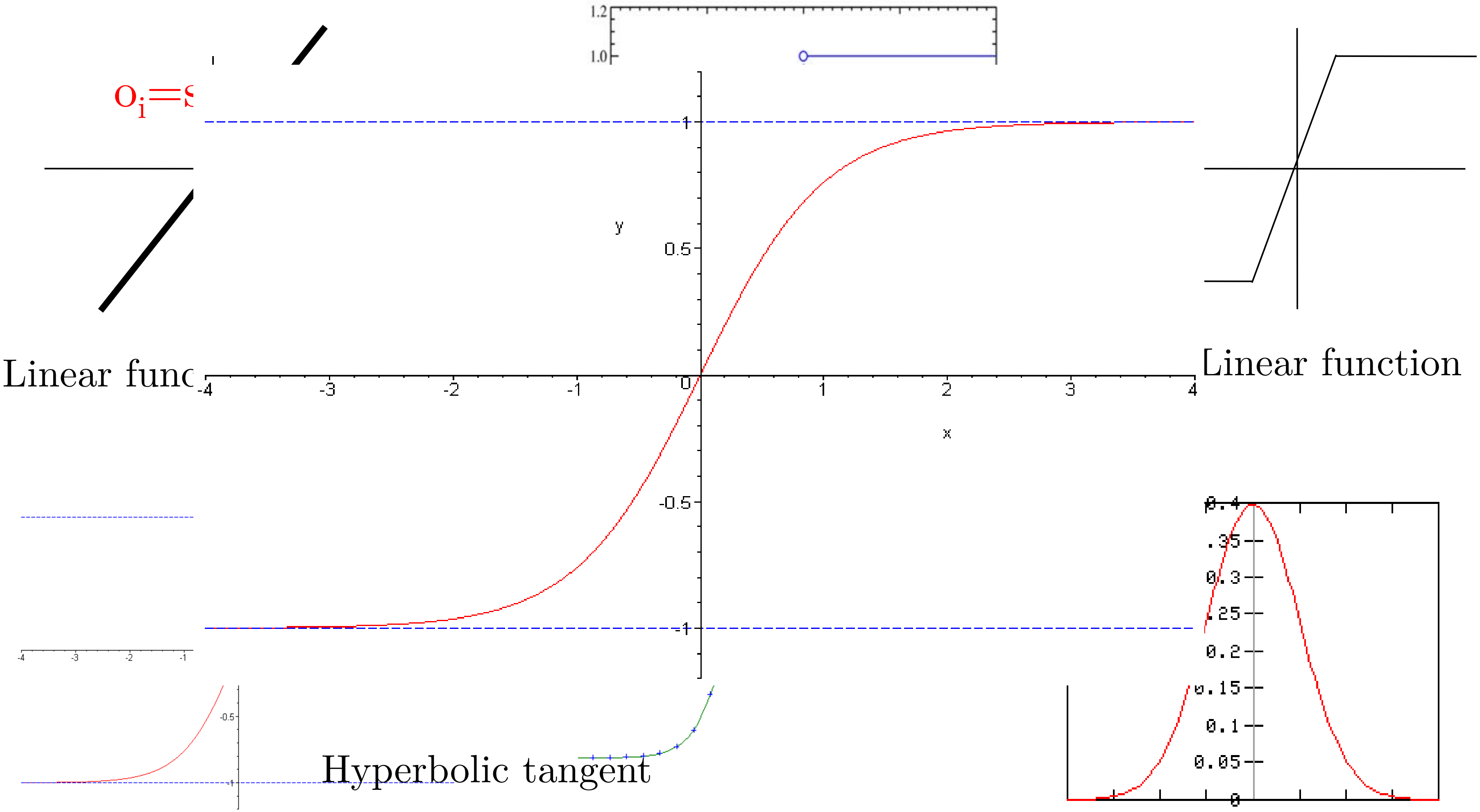


Sigmoïd function



Gaussian function

Differents types of activation function



Linear func

Linear function

Hyperbolic tangent

Sigmoïd function

Hyperbolic tangent

Gaussian function

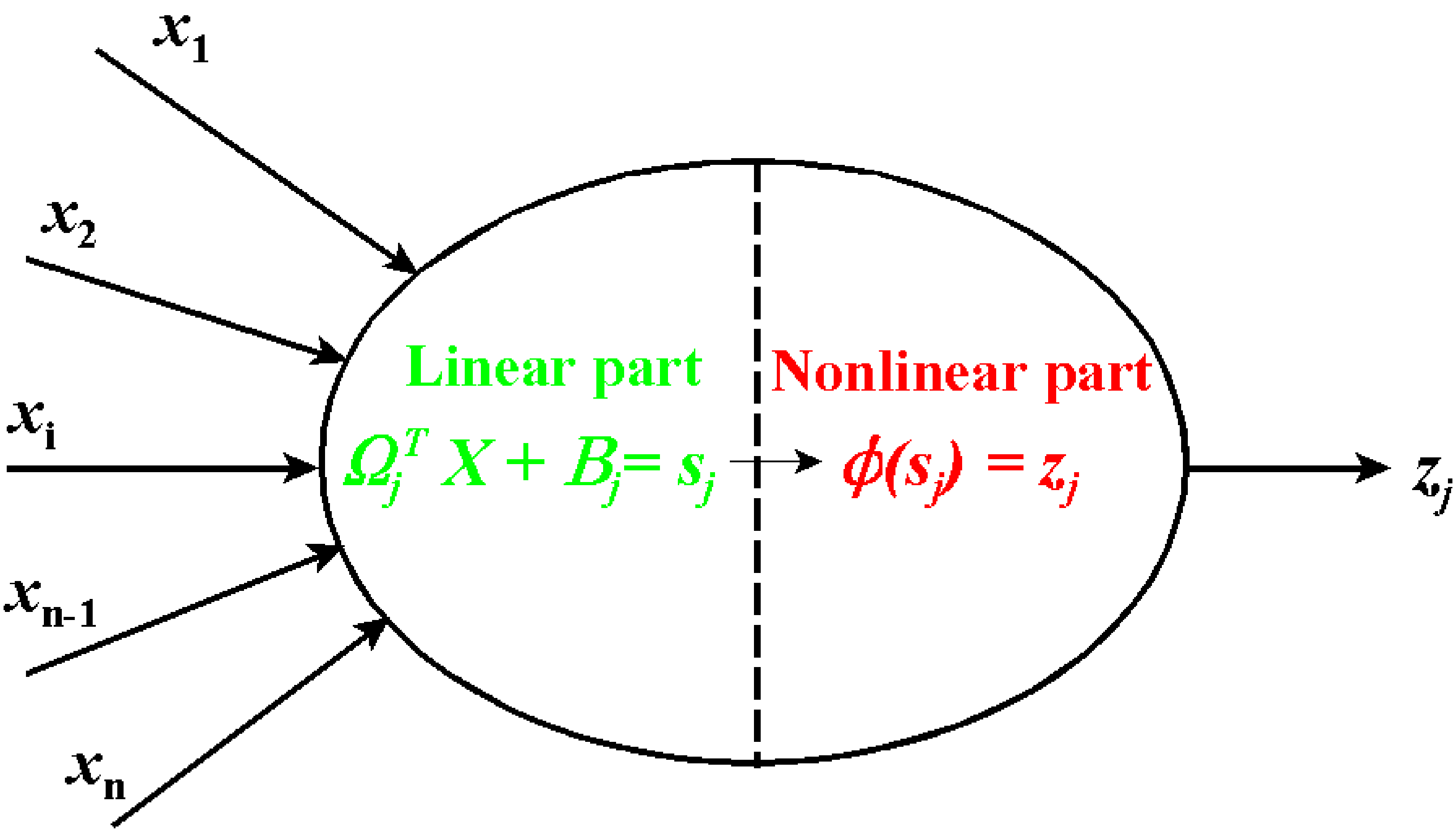
Sigmoïd function

- More used as a activation function
- Introduces non-linearity
- But also a continuous and differentiable function
- Defined by two forms

$$-f(s)=A\frac{e^{Ks}-1}{e^{Ks}+1}=A\tanh(\frac{K}{2}s) \qquad -f(s)=\frac{A}{e^{-Ks}+1}$$

- These functions are bounded
- s → +∝, f(s) → A
- s → - ∝, f(s) → 0
- A regulate the slope of the curve
- Usually, ***A*=1.715095** and **K=4/3**

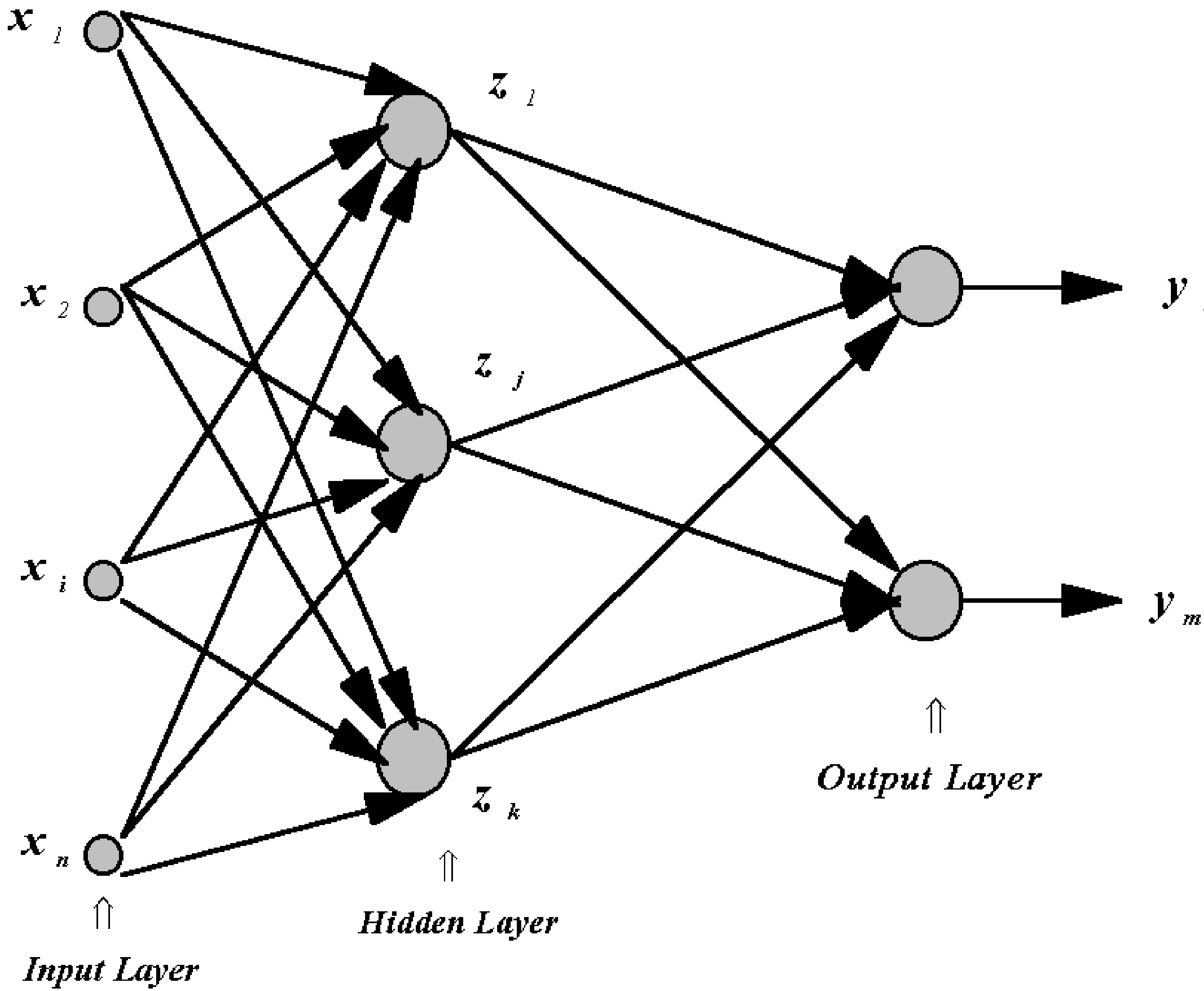
→ linear behaviour of the function between -1 and 1



$$z_j = \phi\left(B_j + \sum_{i=1}^{n} \Omega_{ji} x_i\right) = \boldsymbol{tanh}\left(B_j + \sum_{i=1}^{n} \Omega_{ji} x_i\right)$$

Multi-Layer Perceptron

- Neural Networks organized in successive layers of processing units
- **The neurons are connected by weights and output signals which are function of the sum of the inputs to the neuron modified by the simple nonlinear transfer function**
- Connections running from every unit in one layer to every unit in the next layer
- No others connections permitted
- MLP know as a ***feed-forward neural network***


x_1
x_2
x_i
x_n
z_1
z_j
z_k
y_1
y_m
⇑
Input Layer
⇑
Hidden Layer
⇑
Output Layer

Muli-Layer Perceptron: Definitions

- One input layer corresponding to the inputs of the model $x_1,x_2,\ldots,x_n$
- One output layer corresponding to the ouputs of the model $y_1,y_2,\ldots.,y_n$
- One or more hidden layers
- The numbers of neurons in the hidden layers are not known → Learning phase to determine the numbers and the values of the connexions between the layers (weights)
- $x \rightarrow y=F(x,W)$

Multi-Layer Perceptron: Proprieties

- Multilayer feedforward networks are **universal approximators** (Hornik, 1989)
- NN can be trained to **approximate virtually any smooth, measurable function**
- Can model *highly non-linear functions*
- Can be trained to accurately generalise when presented to new, unseen data

Multi-Layer Perceptron: Training phase

- Training requires a set of training data
- Consists of a series of input x and the corresponding outputs y vectors
- During training, MLP is repeatedly presented with the training data
- The weights of the network are adjusted until the desired input-output mapping occurs
- Training phase is a supervised training
- An error signal is defined as the difference between the desired and the NN-calculated output

Multi-Layer Perceptron: Training phase

- Requirement of a pre-processing before training phase
- Several reasons
  - Inputs (outputs also) of the model can have different orders of magnitude (non-homogeneity of the ranges)
  - Data ranges are often outside the validity bounds of the activation function (usually between -1 and 1)
- Rescaling of the data: $x_i^N = \frac{2}{3} * \left(\frac{x_i - \overline{x_i}}{\sigma(x_i)} \right)$
- Factor 2/3 allows to rescale 80% of the data between -1 and 1

Multi-Layer Perceptron: Training phase

- Choice of the training phase algorithm: **Back-propagation algorithm** (Rumelhart et al., 1986; Bishop, 1994; Haykin, 1996)
- ***Training algorithm consists to determine the weights during an iterative process***:
  - $\mathbf{w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij}(t)}$ where $w_{ij}(t+1)$ is the weight which relies the j-th neuron to the i-th neuron at the time t+1. This value depends to the value of the weight at t
  - Initialisation of the weight for t=0

→ Usually, random initialisation between -1 and 1

Multi-Layer Perceptron: Back-Propagation algorithm

- **Objective** of the training phase is to *find a combination of weights which results in the smallest error*
- The backpropagation training algorithm uses a gradient descent procedure to attempt to locate the absolute (or global) minimum of the error (or cost) function

→ Minimization of a cost function, often taken a the Euclidean distance (or mean-square error) between the desired and NN-calculated NN

- But the choice of the cost function (called ***J***) is a strategic task as it has to take into account all the informations the user has on the inputs and outputs

Calculation of the gradient of the cost function related to the control variables

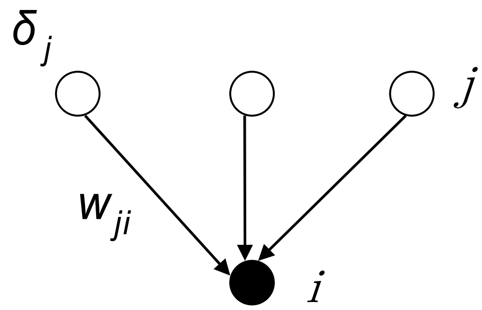
$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial F} \cdot \frac{\partial F}{\partial x}$$

- *Back-Propagation gradient algorithm: Calculation of the partial derivatives of the function F(**x**,W) related to the control variables*

  - Related to the weights of the network w_{ij} ,

  - Related to the input neurons and in particular related to the inputs of the NN (control variables x)

$$\frac{\partial F(x,W)}{\partial x_i} = \sum_j \delta_j w_{ji} \quad \textbf{avec} \quad \delta_j = \frac{\partial J}{\partial s_i}$$



  - F(**x**,**W**): function created by the NN
  - **W**: weight of the connection between i-th neuron and the j-th neuron
  - J: cost function
  - s: state of the i-th neuron

Back-Propagation of the gradient

- $\frac{\partial J}{\partial s_i} = \delta_i$ can be recursively calculated by back-propagation from the output layer:

  – If the index k characterizes a neuron of the output layer,

$$\delta_k = f_k^{'}(s_k).\frac{\partial J}{\partial o_k}$$

  – If i is the index of a hidden neuron, by noting k the index of the neurons which takes the information from the i-th neuron:

$$\delta_i = f_i^{'}(s_i).\sum_k \delta_k.w_{ki}$$

  – If i is the index of a neuron of the input layer; f is the identity function:

$$\frac{\partial J}{\partial x} = \sum_i \delta_i w_{ij}$$

• For a given architecture of the NN, only the knowledge of the derivatives of the cost function **related to the states of the output neurons** dJ/do occurs in the calculation of the gradient

Multi-Layer Perceptron: Back-Propagation algorithm

- The backpropagation algorithm can be summarized as followed:

  1) Initialise network weights
  2) Present first input vector, from training data, to the network
  3) Propagate the input vector through the network to obtain an output
  4) Calculate an error signal by comparing actual output to the desired (target) output
  5) Adjust weights to minimise overall error
  6) Repeat 2-7 with the next input vector, until overall error is satisfactorily small

Multi-Layer Perceptron: Back-Propagation algorithm

- Two main kind of training
  - **sequential training**: the network weights are adapted after each pattern has been presented
  - **batch training**: the summed error for all patterns is used to update the weights
- Backpropagation algorithm contains **two adjustable parameters:**
  - learning rate: determine the step size taken during the iterative gradient descent learning process
  - momentum term: used to assist the gradient descent process if it becomes stuck in a local minimum

Multi-Layer Perceptron: Test phase

- Optimal architecture of the NN is obtained by doing successive tests in which the number of neurons and hidden layers are increasing, using the validation set
- **Optimal architecture**=**minimal error for minimal number of neurons**
- When training phase finished, the NN model does algebraic operations only

Multi-Layer Perceptron: Test phase

- Statistical parameters to define the performances of the NN:
  - Root-Mean-Square Error (RMS)
  - Relative error
  - Correlation coefficient

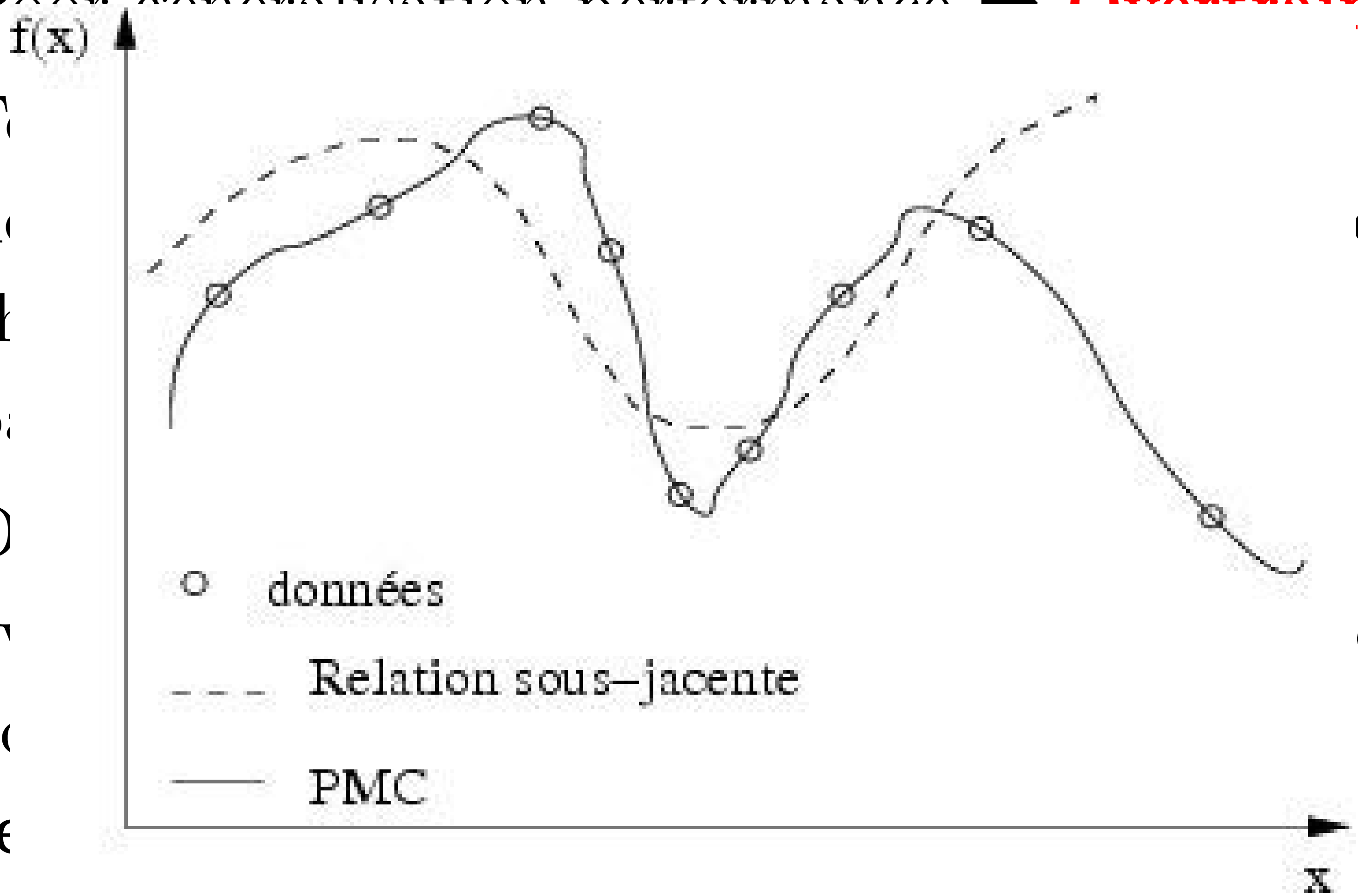
Multi-Layer Perceptron: Overtraining

- Poor generalisation performance → **Overtraining**
- For instance, given some training data and a network with too many nodes and hidden layers, the network will eventually learn all the training patterns in the training data
- One way to ensure that the network is not overtrained and that the generalisation will be good, is to **divide the training dataset into several sets**: training, validation and test

→ **Cross-validation**

Multi-Layer Perceptron: Overtraining

- Poor generalisation performance → Overtraining



- F… n… ers, th… ing p…
- O… o… e g… se…

→ Cross-validation

Multi-Layer Perceptron: Overtraining

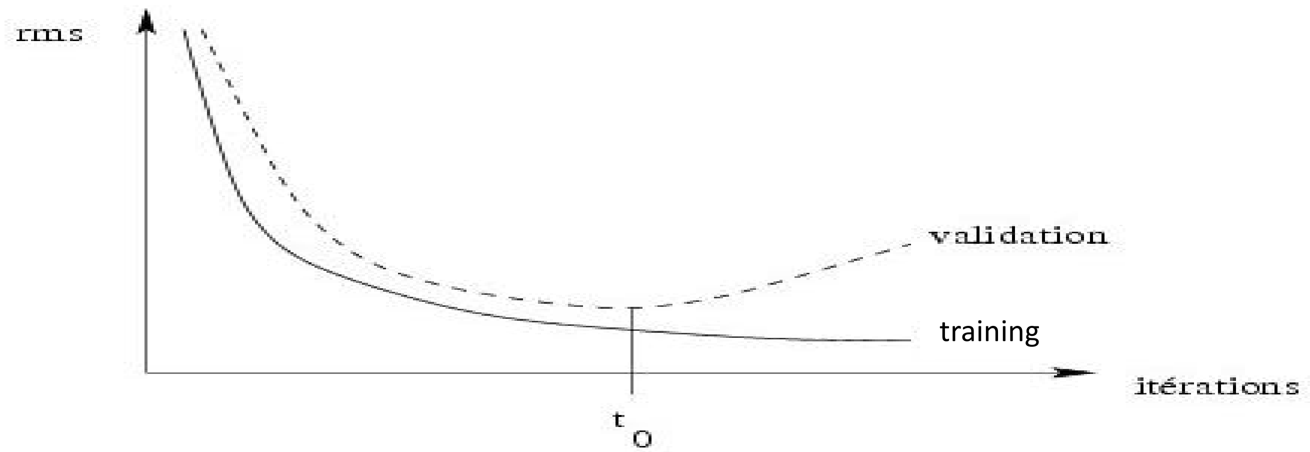
- Training dataset: used to train the network
- Validation dataset: used to assess the generalisation ability of the network whilst training is occurring
- Test set: used to assess the generalisation performance of the NN
- Usually number of NN parameters << number of observations in the dataset → Not always attainable in environmental problems

Multi-Layer Perceptron: Test phase

- Training stopped when the generalisation performance reaches a max

- When overtraining occurs, the training error keep decreasing while the validation error starts to increase.

→The technique is known as **early stopping**



The architecture is chosen by comparing the validation error

Using NN

- Can be used as a « black-box » but warning !!!
- Matlab toolbox
- Python toolbox:
  - https://keras.io/
  - https://scikit-learn.org/stable/

Applications

- Vertical attenuation coefficient, K_d
- Atmospheric correction of ocean color images
- Estimation of the Inherent Optical Properties of the seawater
- Estimation of the chlorophyll-a concentration
- Estimation of nutrients

Applications

- **Vertical attenuation coefficient, K_d**
- Atmospheric correction of ocean color images
- Estimation of the Inherent Optical Properties of the seawater
- Estimation of the chlorophyll-a concentration
- Estimation of nutrients

Diffuse attenuation coefficient K_d

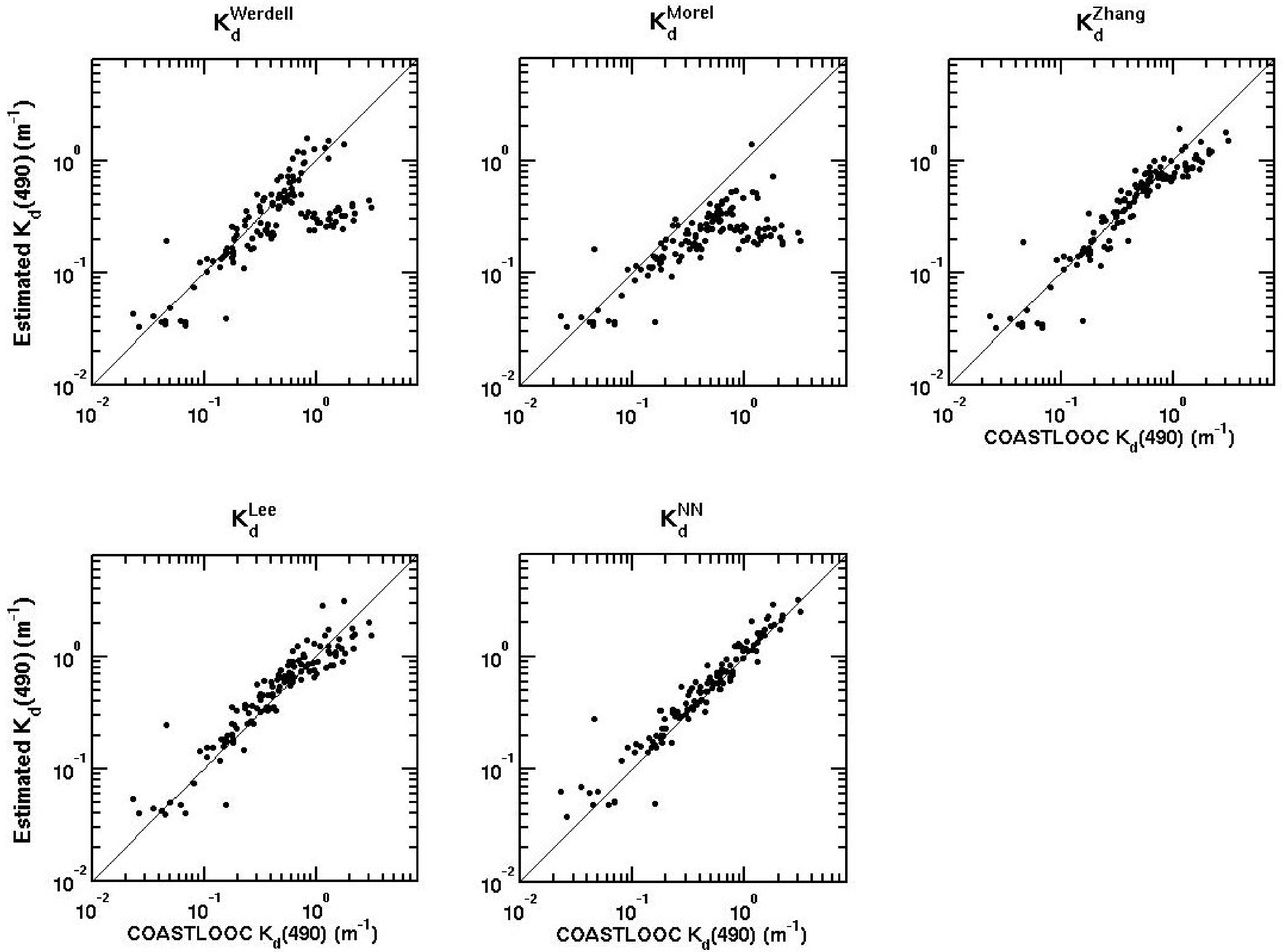
- Diffuse attenuation coefficient of downwelling irradiance $K_d(\lambda)$:

$$K_d(z) = \frac{1}{E_d(z)} \frac{d\,E_d(z)}{dz}$$

- **Critical role in the regulation of biological and physical processes**
  - Heat transfer (Lewis et al., 1990; Morel and Antoine, 1994)
  - Photosynthesis (Platt et al., 1988)
  - Visibility (Preisendorfer, 1986)

Validation in the European waters

- **Development of the method**
  - Artificial Neural Networks
  - Simulated + *in-situ* datasets
  - $K_d(490)$: [0.03; 3.5] m^{-1}
  - λ(**SeaWiFS**)=[412, 443, 490, 555, 670]
  - Inputs: spectrum of $R_{rs}(\lambda) + \lambda$
  - Outputs: $K_d(\lambda)$
- **Validation**
  - Data from COASTLOOC (Babin et al., 2003)
  - European Waters
  - Values of Kd between 412 and 667 nm



$K_d^{Werdell}$
K_d^{Morel}
K_d^{Zhang}
K_d^{Lee}
K_d^{NN}
Estimated $K_d(490)$ (m^{-1})
COASTLOOC $K_d(490)$ (m^{-1})
10^{-2}
10^{-1}
10^{0}

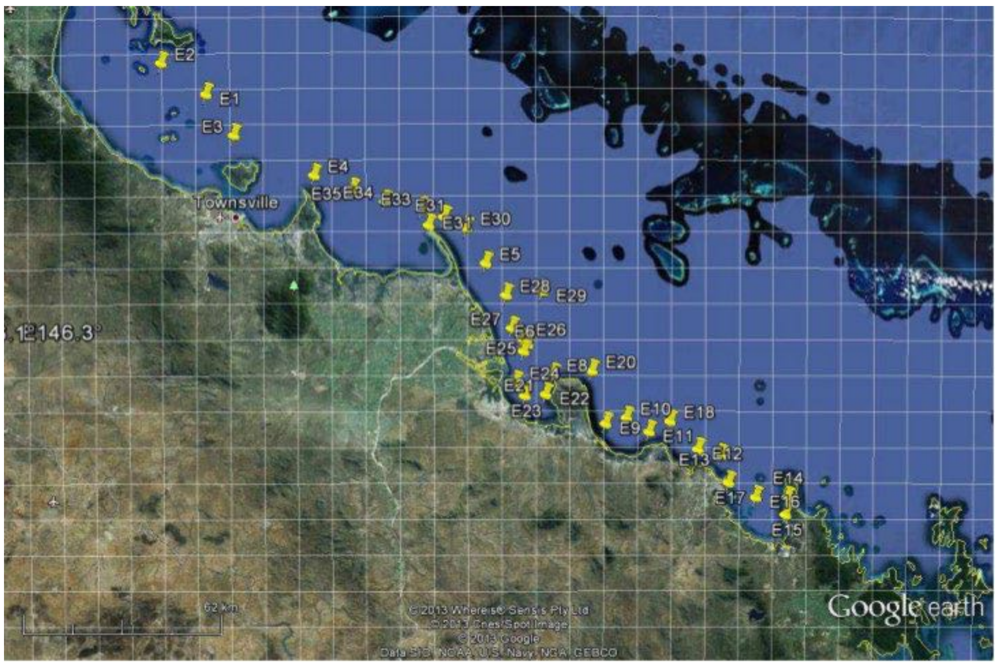
in-situ measurements in the Great Barrier Reef (Australia):

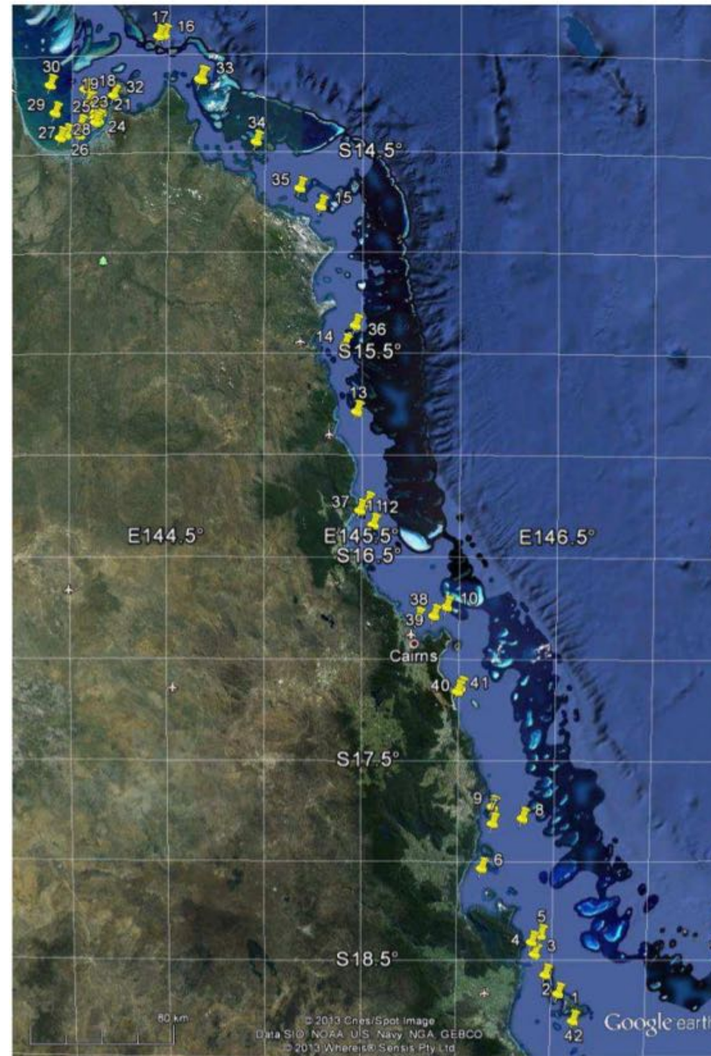
- 186 points of validation
- $K_d(490)$: [0.166; 1.90] m^{-1}, mean=0.411 m^{-1}; std=0.33 m^{-1}
- Burkedin River, Moreton Bay, Sud de la Nouvelle Galles

Jamet et al., in prep

in-situ measurements in the Great Barrier Reef (Australia):

- 186 points of validation
- $K_d(490)$: [0.166; 1.90] m^{-1}, mean=0.411 m^{-1}; std=0.33 m^{-1}
- Burkedin River, Moreton Bay, Sud de la Nouvelle Galles





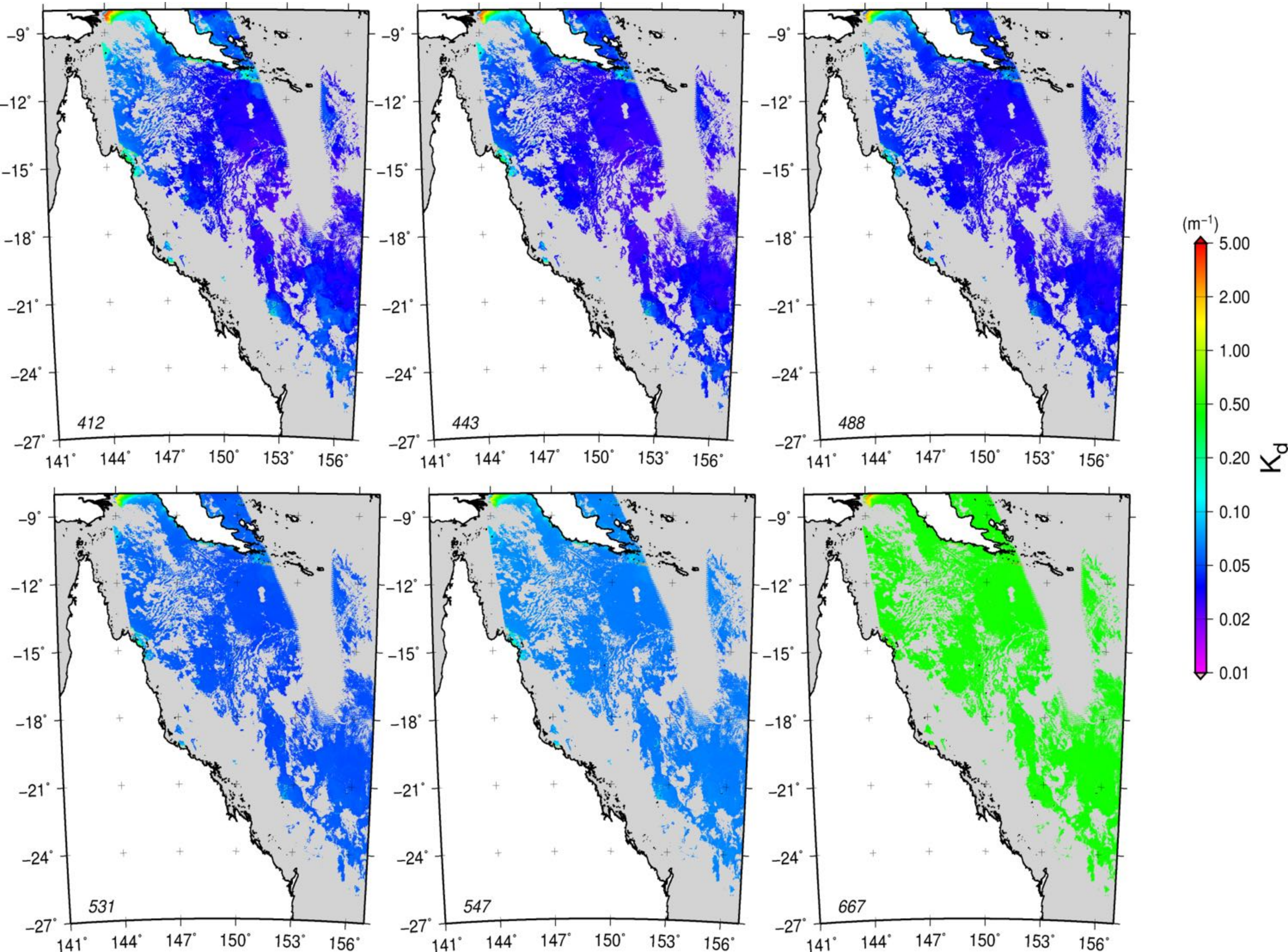
in-situ measurements in the Great Barrier Reef (Australia):

- 186 points of validation
- $K_d(490)$: [0.166; 1.90] m^{-1}, mean=0.411 m^{-1}; std=0.33 m^{-1}
- Burkedin River, Moreton Bay, Sud de la Nouvelle Galles

TAB. 4.1 – *Résultats statistiques pour l'évaluation de différents algorithmes estimant $K_d(490)$ dans la Grande Barrière de Corail*

	K_d^{NASA}	K_d^{Morel}	K_d^{Zhang}	K_d^{Tiwari}	K_d^{Wang}	K_d^{NN}
RMS	34.03	0.378	0.763	0.251	0.767	**0.199**
Relative Error (%)	278	55	55	58	57	**51**
ADP	1.57	1.34	0.94	1.20	0.93	**0.81**
Slope	0.002	0.44	0.23	0.63	0.21	**0.76**
Intercept	0.34	0.22	0.26	0.17	0.26	**0.12**
Correlation coefficient (%)	30	73	69	81	60	**84**
bias	3	-0.05	0.03	-0.067	0.08	**-0.046**
% in ± 25 %	33	22	41	30	43	**47**

Jamet et al., in prep

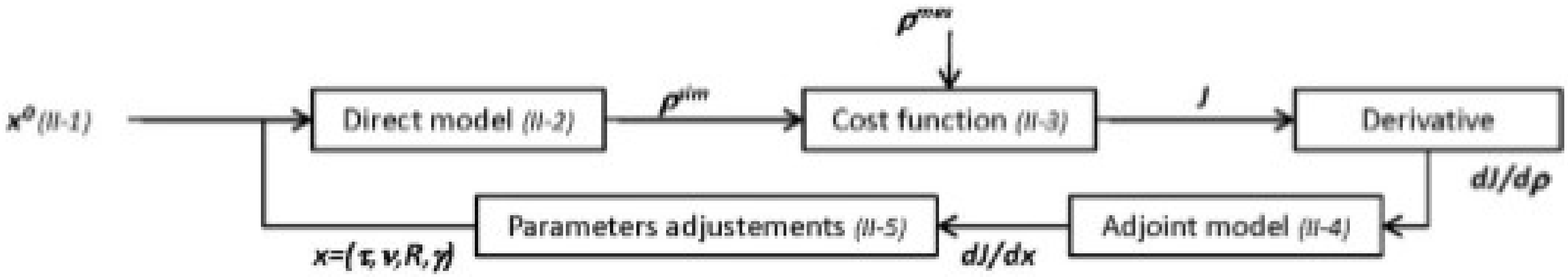

412
443
488
531
547
667
-9°
-12°
-15°
-18°
-21°
-24°
-27°
141°
144°
147°
150°
153°
156°
(m⁻¹)
5.00
2.00
1.00
0.50
0.20
0.10
0.05
0.02
0.01
K_d

Applications

- **Atmospheric correction of ocean color images**
- Estimation of the Inherent Optical Properties of the seawater
- Estimation of the chlorophyll-a concentration
- Estimation of nutrients

Spectral matching/optimization algorithm

- Chomko et al. (1998); Kuchinke et al. (2009)
- Jamet et al. (2004); Brajard et al. (2010)
- Li et al. (2003); Stamnes et al. (2005)



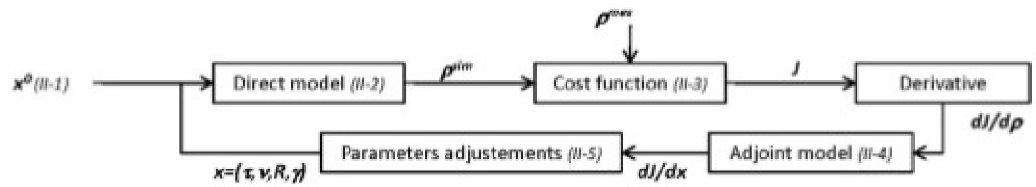
- Ocean and atmosphere are coupled
- Use of LUT or NN or ... for simulating (La+Lra, t, Lw)
- Allow to deal with absorbing aerosols/coastal waters

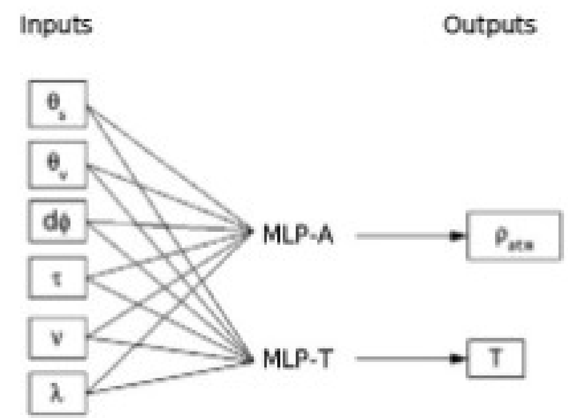
Neuro-variational inversion of ocean color images

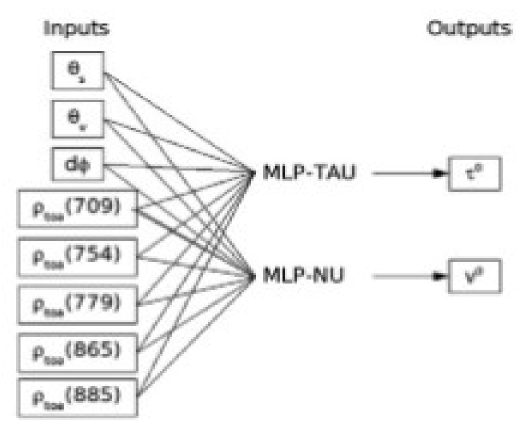
- Joint MLP and Variational inversion
- MLP: Direct modelling from oceanic and atmospheric parameters
- Variational inversion: Iterative modification of the oceanic and atmospheric parameters to model the top of the atmosphere signal using a cost function

$$J(\tau,\upsilon,R,\gamma)=\sum_{i=1}^{5}s_i\left[\rho_{toa}^{mes}(\lambda_i)-\rho_{toa}^{sim}(\lambda_i,\tau,\upsilon,R,\gamma)\right]^2+\beta^{\tau}\left(\tau-\tau^0\right)^2$$
$$+\beta^{\upsilon}\left(\upsilon-\upsilon^0\right)^2+\beta^{R}\left(R-R^0\right)^2+\beta^{\gamma}\left(\gamma-\gamma^0\right)^2$$

- Need of a first guess → Use of MLP for a first approximation of the atmospheric parameters







Brajard et al., 2006a, 2006b, 2012

Table 4
Description of MLP-TAU and MLP-NU.

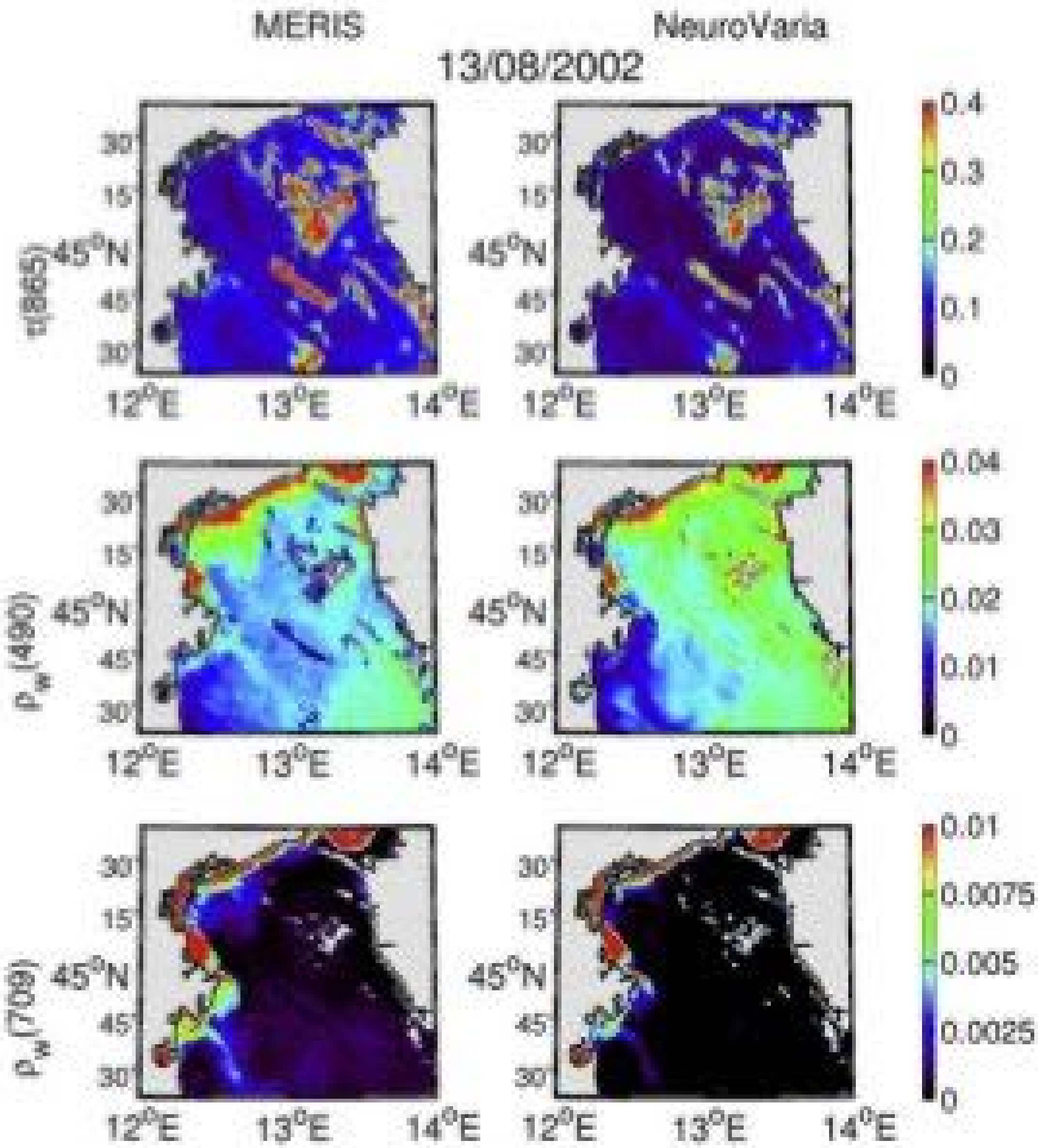
MLP	MLP-TAU	MLP-NU
Inputs (8 neurons)	$\rho_{toa}(709,754,779,865,885),\theta_s,\theta_v,d\varphi$	
Output (1 neuron)	τ^0	ν^0
Number of hidden layers	2	2
Size of the 1st hidden layer	30	30
Size of the 2nd hidden layer	25	25
RMS[a]	1.58×10^{-2}	3.79×10^{-2}
Relative error[b]	2.77%	0.6%
r^{2c}	1.0	1.0

[a] Root mean square error.
[b] Relative error.
[c] Correlation coefficient.

Table 5
Description of MLP-A and MLP-T.

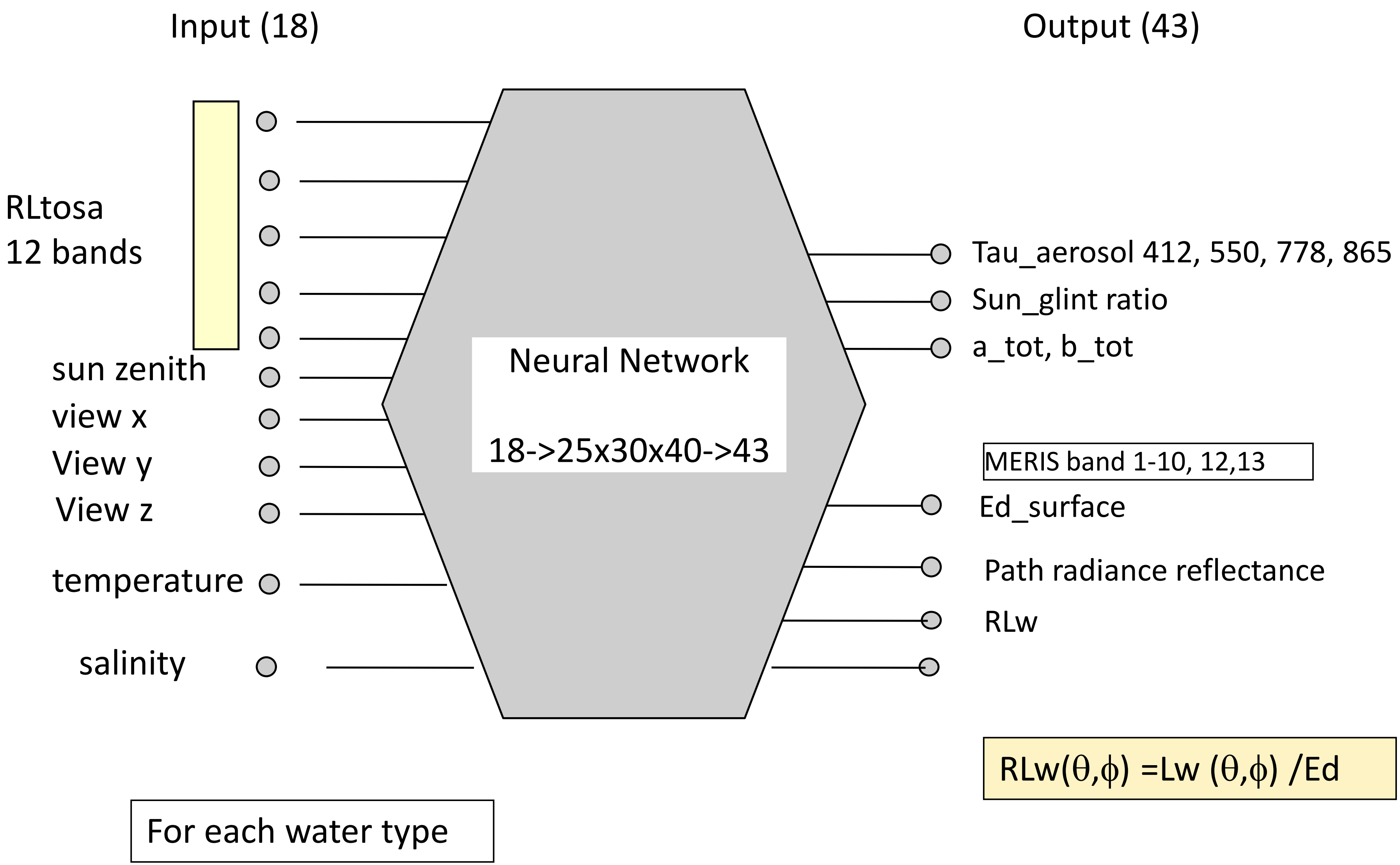
MLP	MLP-A	MLP-T
Inputs	$\theta_s,\theta_v,\Delta\phi,\lambda,\tau,\nu$	θ,λ,τ,ν
Output	ρ_{atm}	T
Number of hidden layers	2	1
Size of the 1st hidden layer	28	16
Size of the 2nd hidden layer	34	–
RMS[a] for $\lambda \geq 708$ nm	1.43×10^{-3}	6.9×10^{-4}
Relative error[b] for $\lambda \geq 708$ nm	1.11%	0.04%
r^{2c}	0.998	1.0
	RMS (rel. err.)	RMS (rel. err.)
$\lambda = 709$ nm	1.4×10^{-3} (1.1%)	6.3×10^{-4} (0.04%)
$\lambda = 754$ nm	1.5×10^{-3} (1.0%)	4.8×10^{-4} (0.04%)
$\lambda = 779$ nm	1.5×10^{-3} (1.1%)	5.2×10^{-4} (0.03%)
$\lambda = 865$ nm	1.2×10^{-3} (1.2%)	9.7×10^{-4} (0.04%)
$\lambda = 885$ nm	1.4×10^{-3} (1.2%)	7.4×10^{-4} (0.04%)
$\lambda = $ *412 nm to 681 nm (visible)*	*1.8×10^{-3} (0.8%)*	*6.9×10^{-4} (0.04%)*

[a] Root mean square error.
[b] Relative error.
[c] Correlation coefficient.

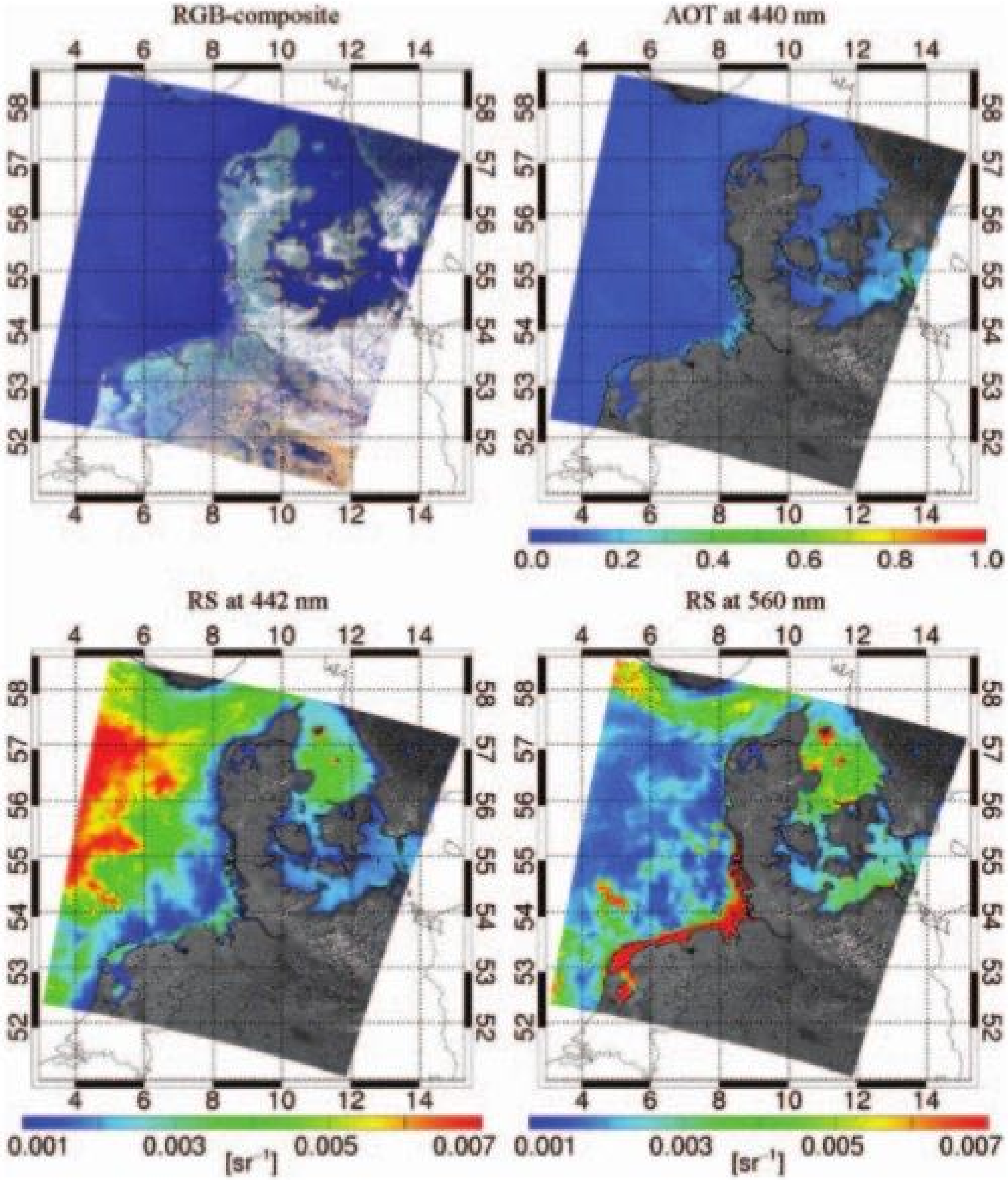


Brajard et al., 2006a, 2006b, 2012

Inverse NN for atmospheric correction



Doerffer and Schiller, 2007


RGB-composite
AOT at 440 nm
RS at 442 nm
RS at 560 nm
0.0 0.2 0.4 0.6 0.8 1.0
0.001 0.003 0.005 0.007
[sr^-1]

Inverse NN for atmospheric correction

Schroeder et al. (2007)

- MLP
- Inversion directe des images L1B vers L2 (TOA L → Rrs)
- Apprentissage: 100000 données
- Test: 100000 données
- Couche d'entrée: 8 images à différentes λ, géométrie d'observation et du Soleil, pression de surface et vitesse du vent
- Couche de sortie: 8 images couleur de l'océan à différentes λ + épaisseur optique aérosol à 4 λ

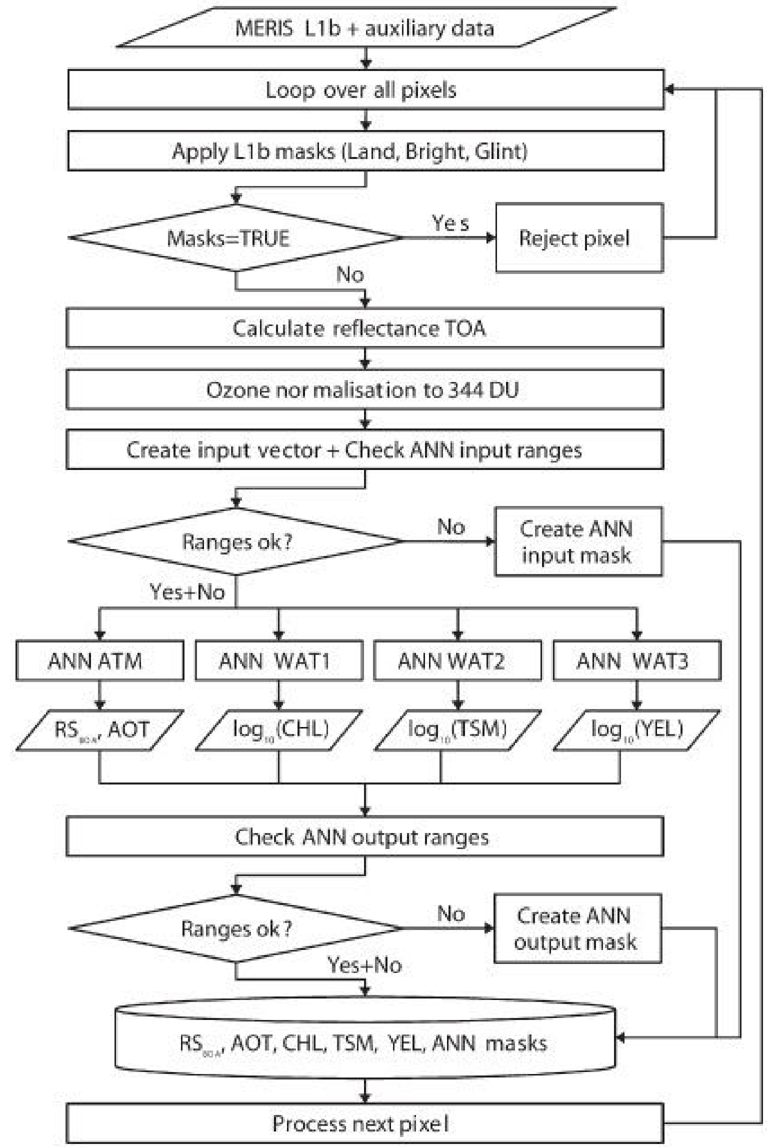
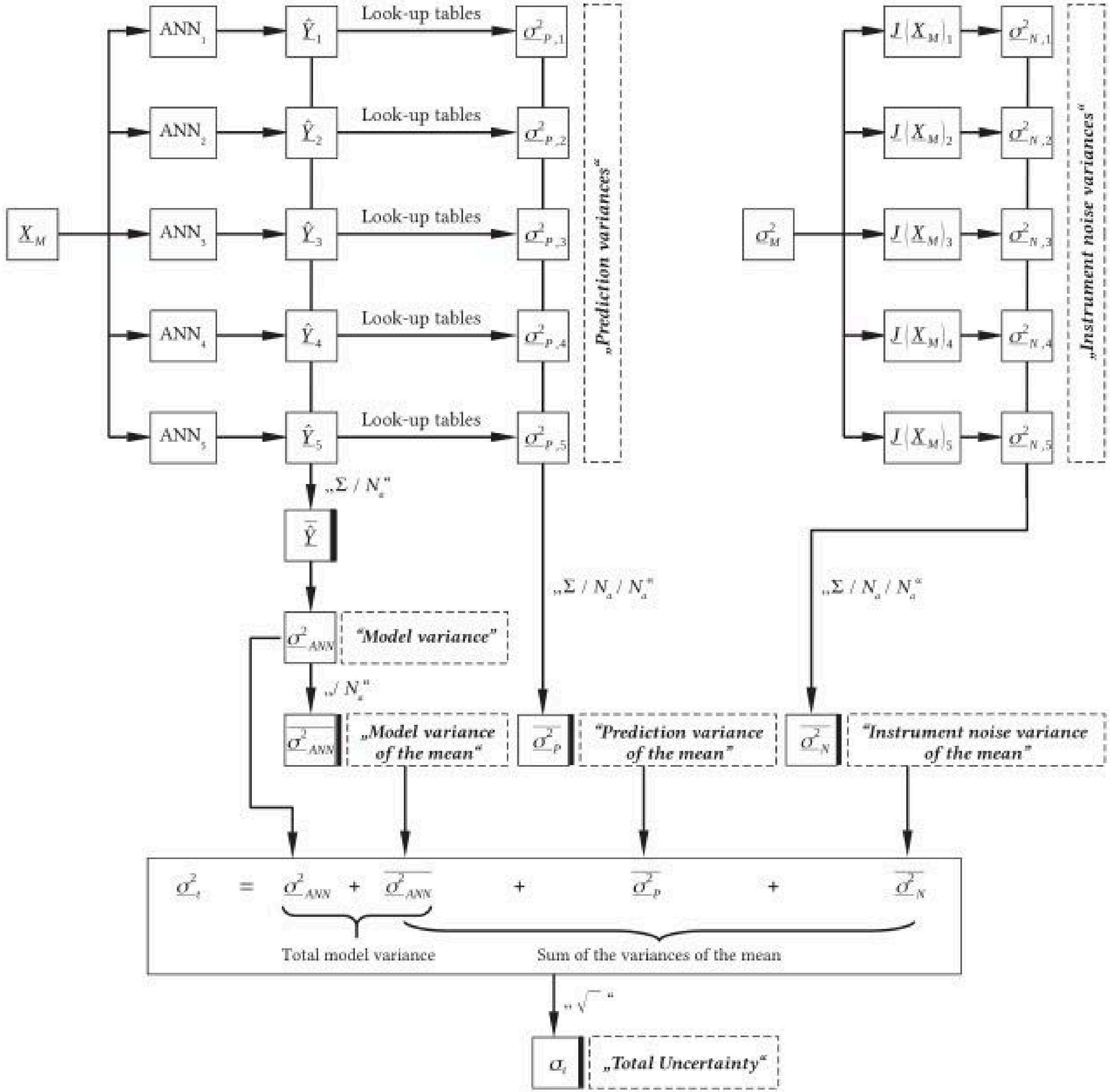
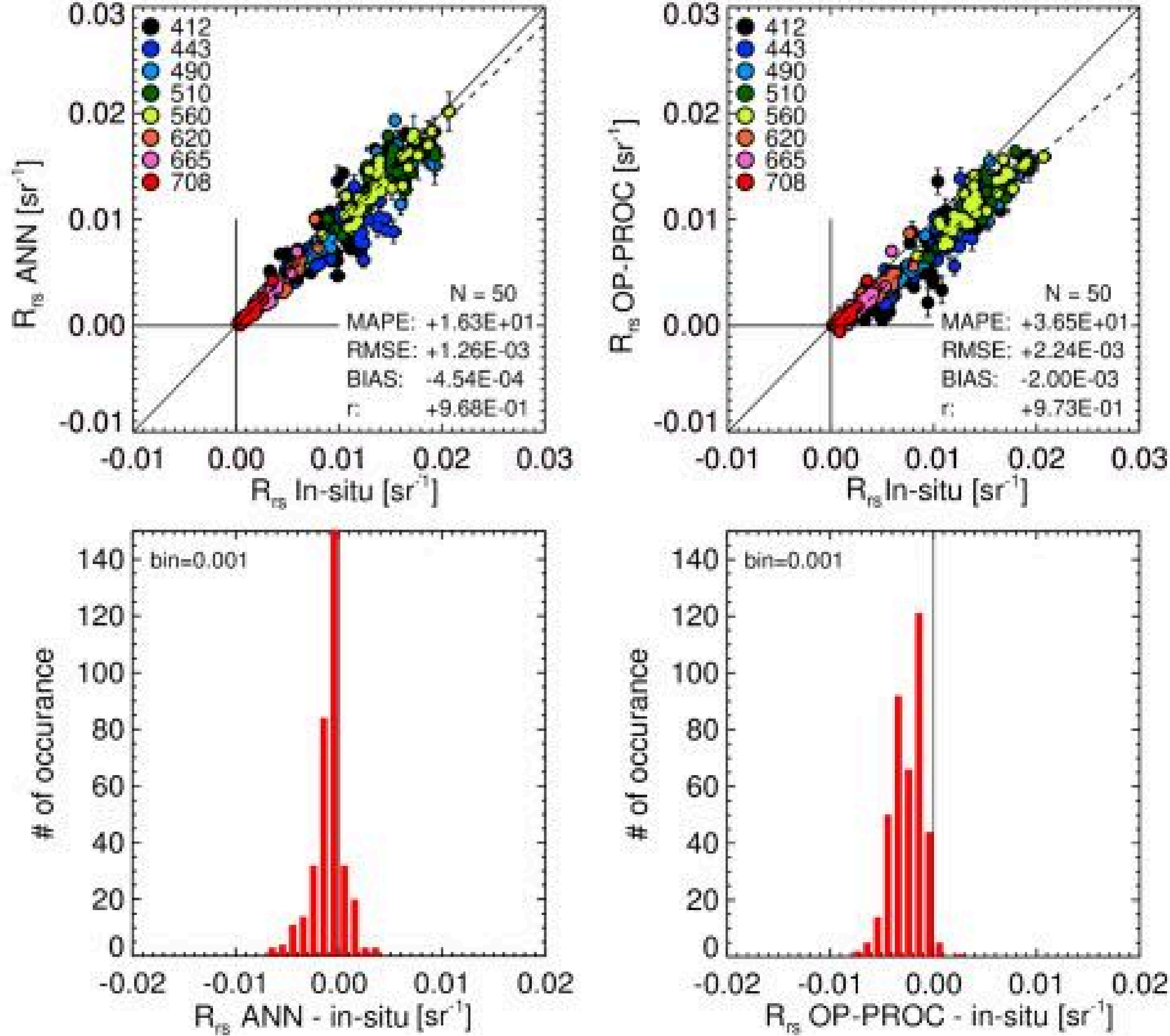


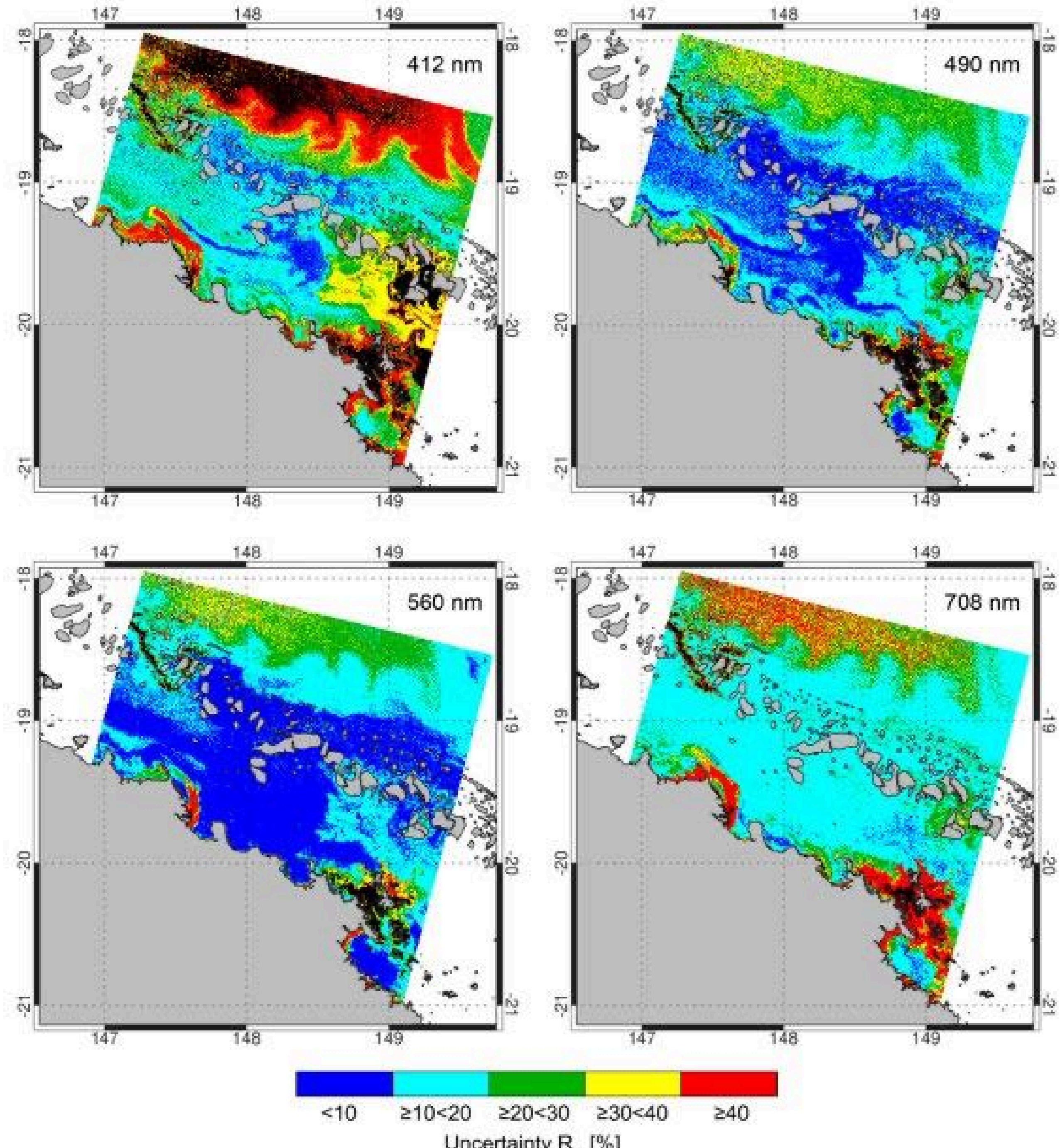
Figure 1. Flowchart of the proposed case-2 water processor.



$\underline{X}_M$
ANN1
ANN2
ANN3
ANN4
ANN5
Look-up tables
Look-up tables
Look-up tables
Look-up tables
Look-up tables
„Prediction variances“
σ^2_M
„Instrument noise variances“
„Σ / Na“
„Σ / Na / Na“
„Σ / Na / Na“
“Model variance”
„/ Na“
„Model variance of the mean“
“Prediction variance of the mean”
“Instrument noise variance of the mean”
Total model variance
Sum of the variances of the mean
„Total Uncertainty“



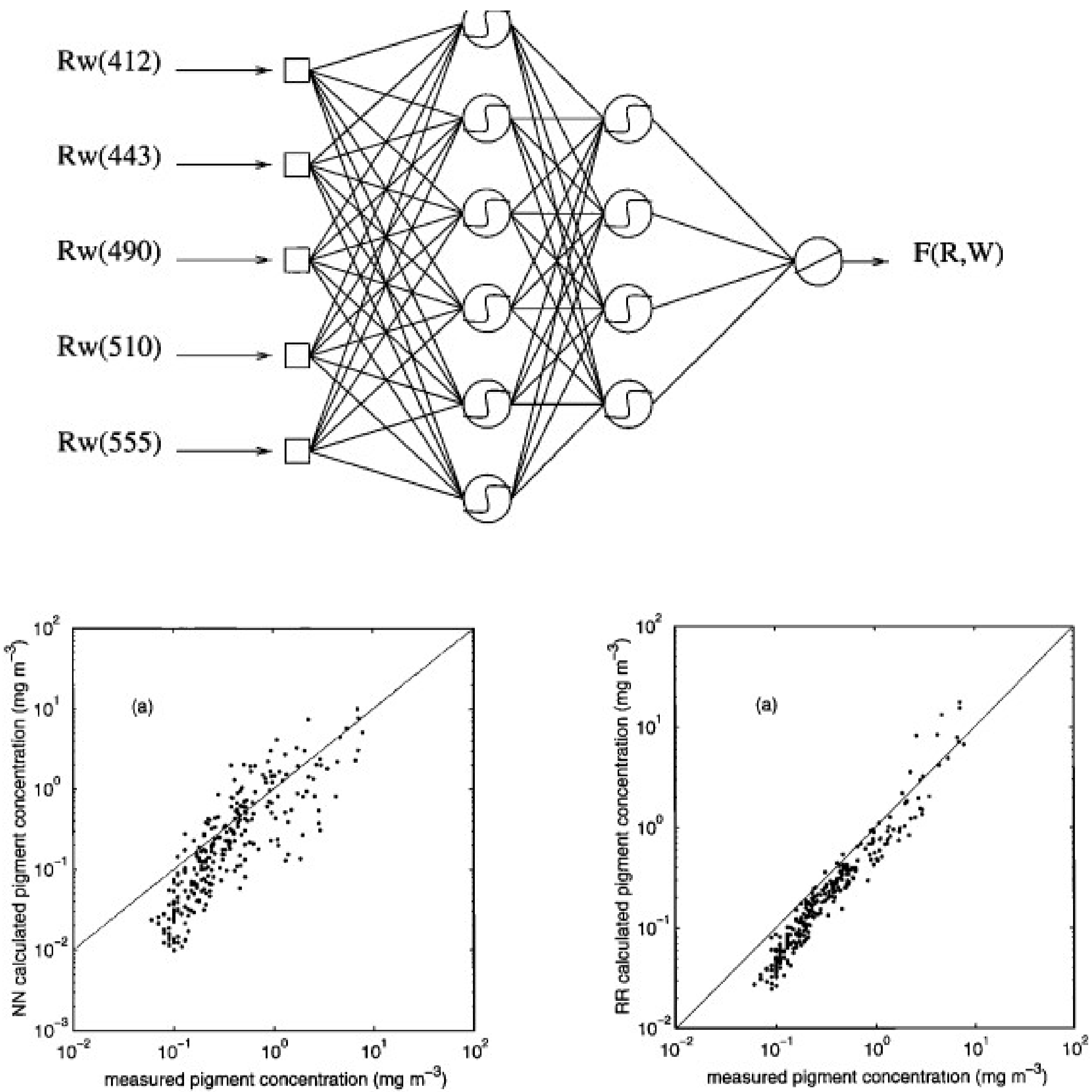
412
443
490
510
560
620
665
708
N = 50
MAPE: +1.63E+01
RMSE: +1.26E-03
BIAS: -4.54E-04
r: +9.68E-01
R_{rs} ANN [sr^{-1}]
R_{rs} In-situ [sr^{-1}]
N = 50
MAPE: +3.65E+01
RMSE: +2.24E-03
BIAS: -2.00E-03
r: +9.73E-01
R_{rs} OP-PROC [sr^{-1}]
R_{rs} In-situ [sr^{-1}]
bin=0.001
# of occurance
R_{rs} ANN - in-situ [sr^{-1}]
bin=0.001
# of occurance
R_{rs} OP-PROC - in-situ [sr^{-1}]



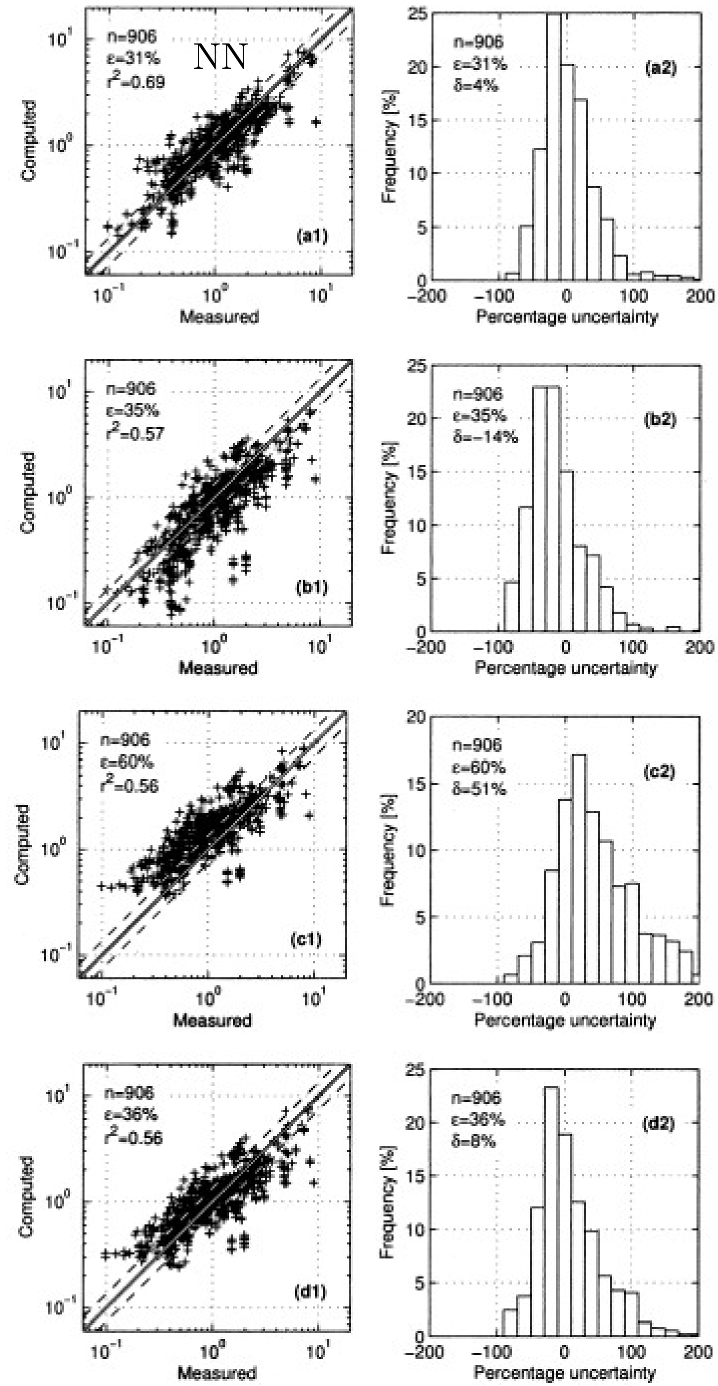
412 nm
490 nm
560 nm
708 nm
147
148
149
-18
-19
-20
-21
<10
≥10<20
≥20<30
≥30<40
≥40
Uncertainty R_rs [%]

Applications

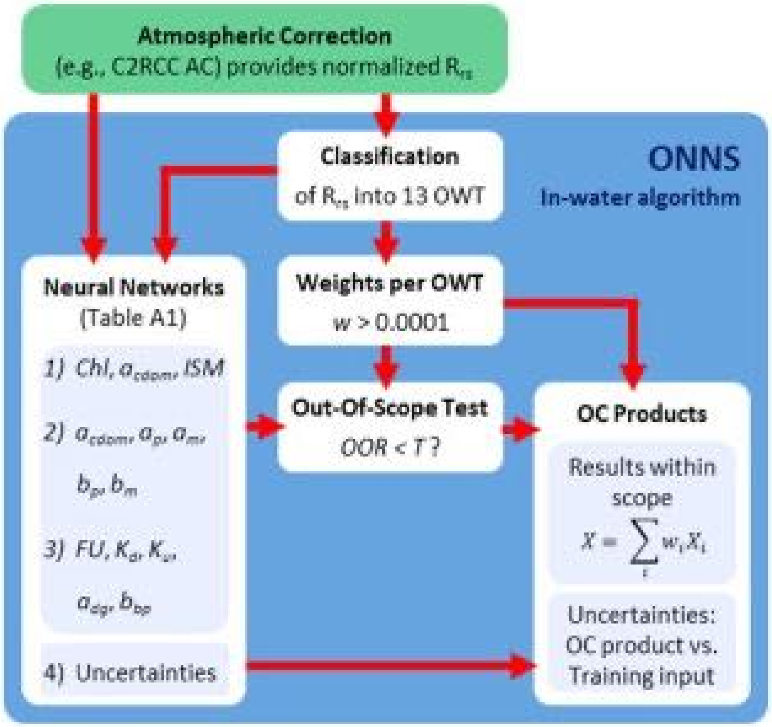
- Atmospheric correction of ocean color images
- **Estimation of the chlorophyll-a concentration**
- Estimation of the Inherent Optical Properties of the seawater
- Estimation of nutrients

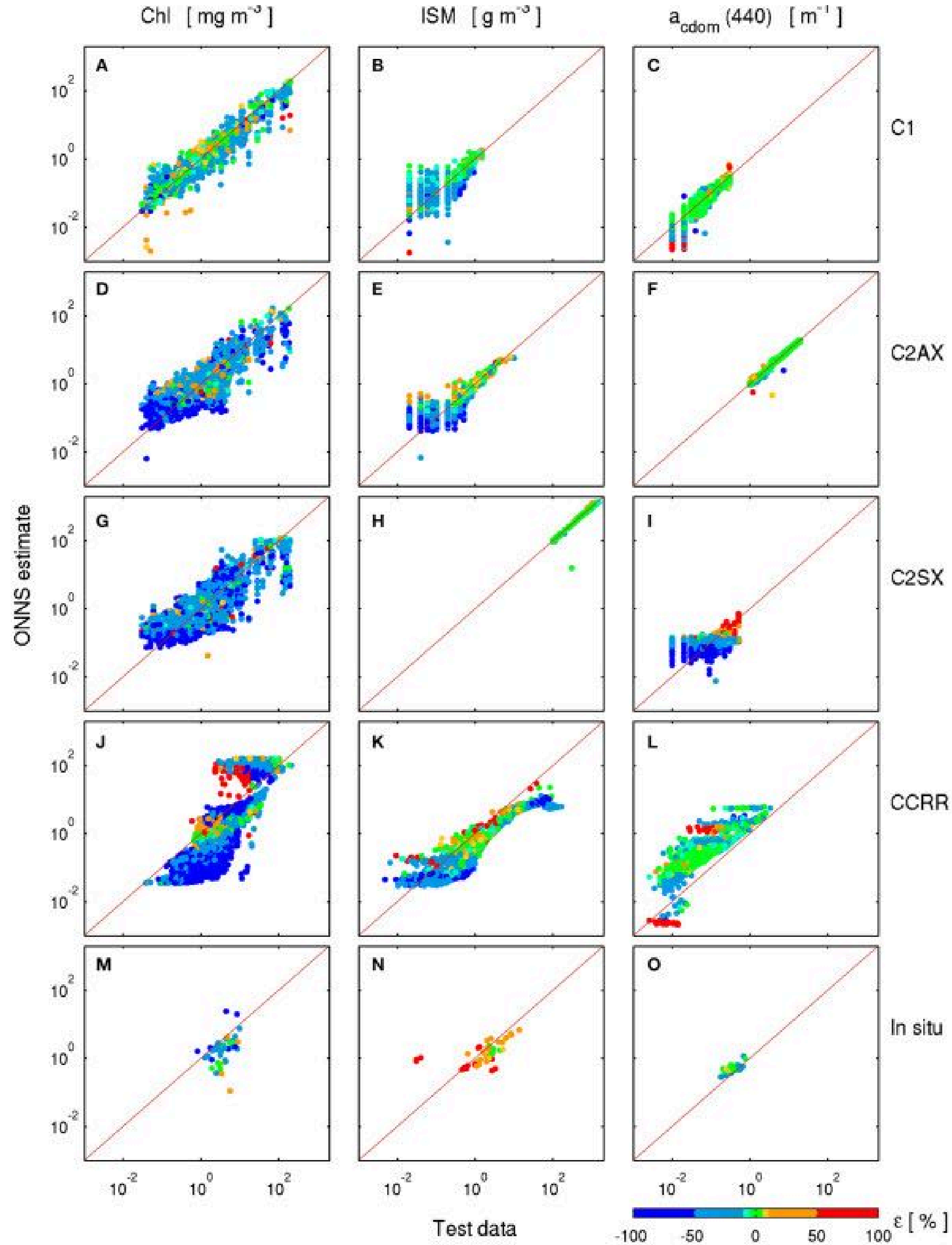


Gross et al. (2000)

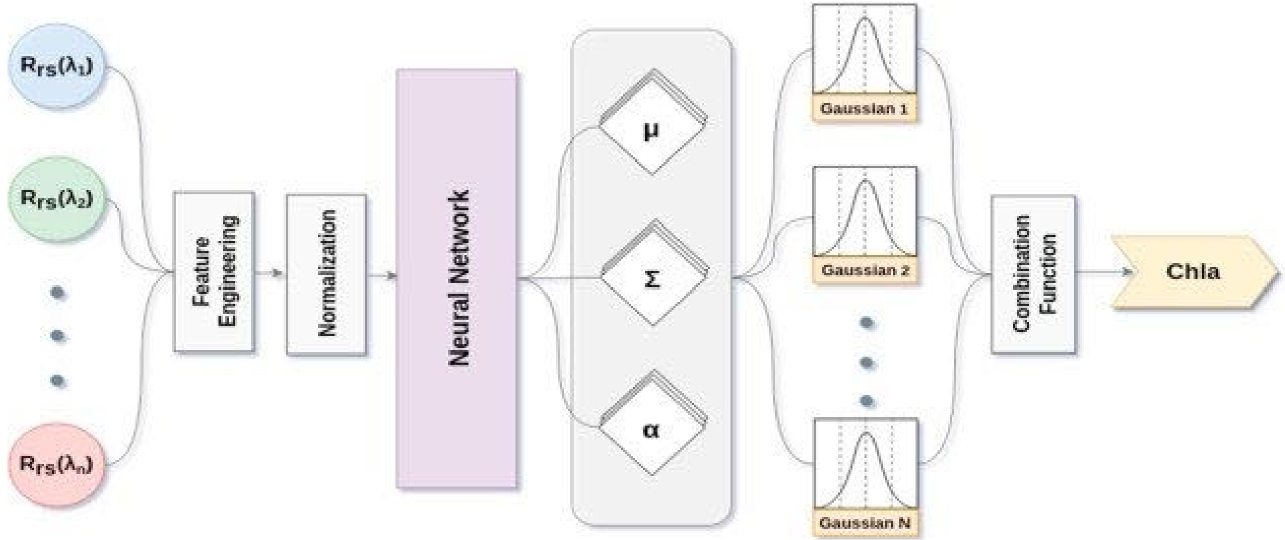


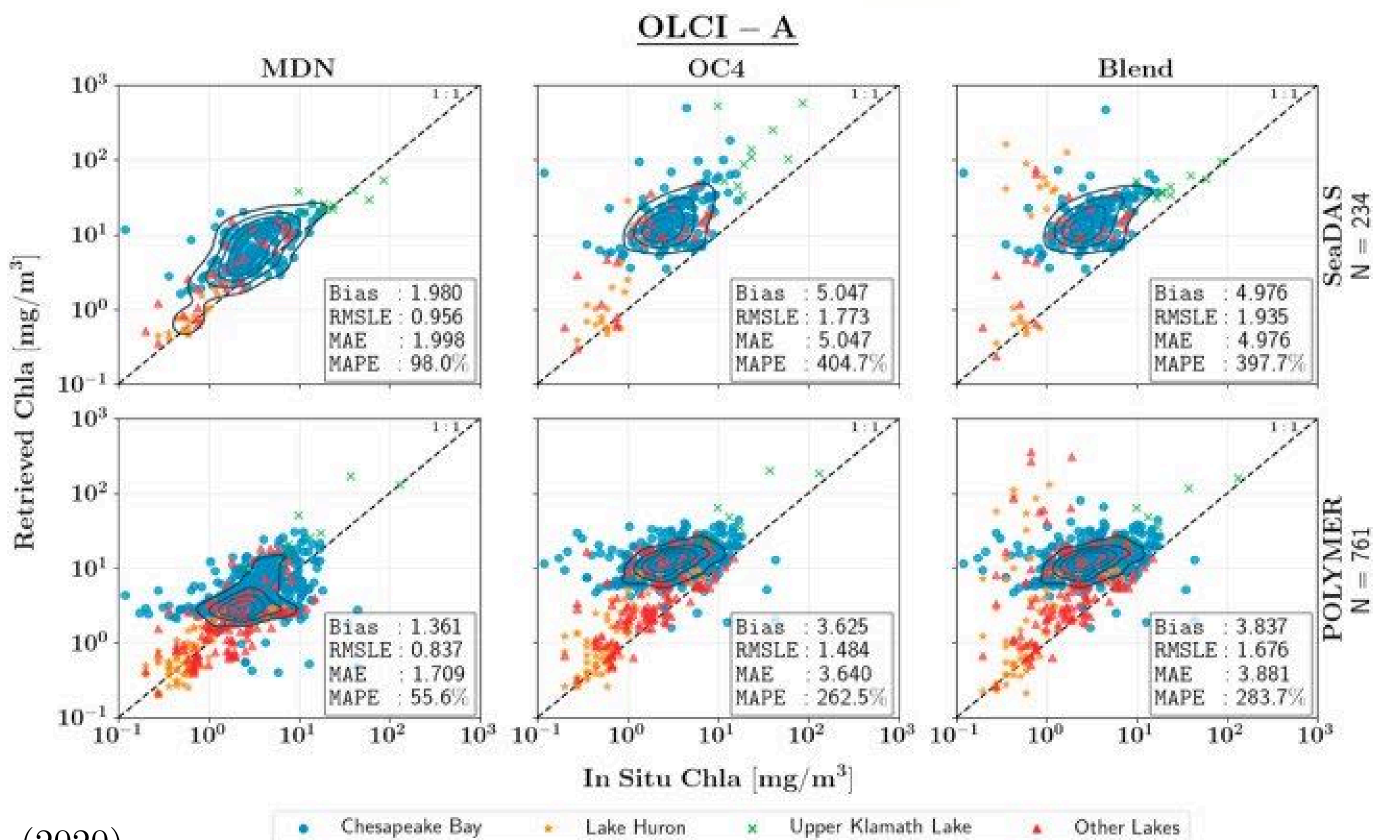
D'Alimonte and Zibordi (2003)





Hieronymi et al. (2020)

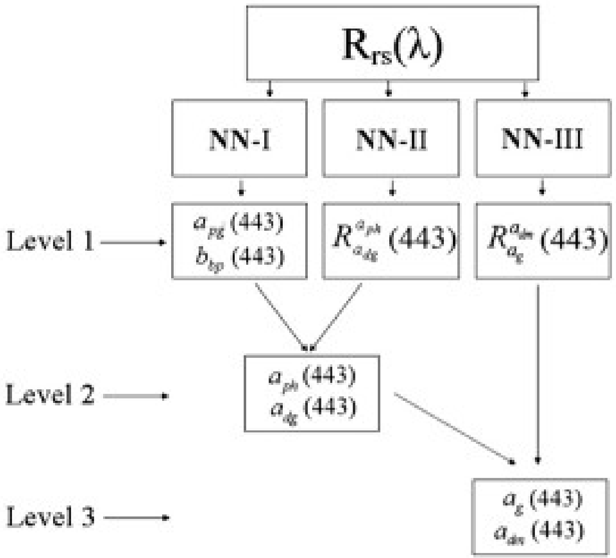


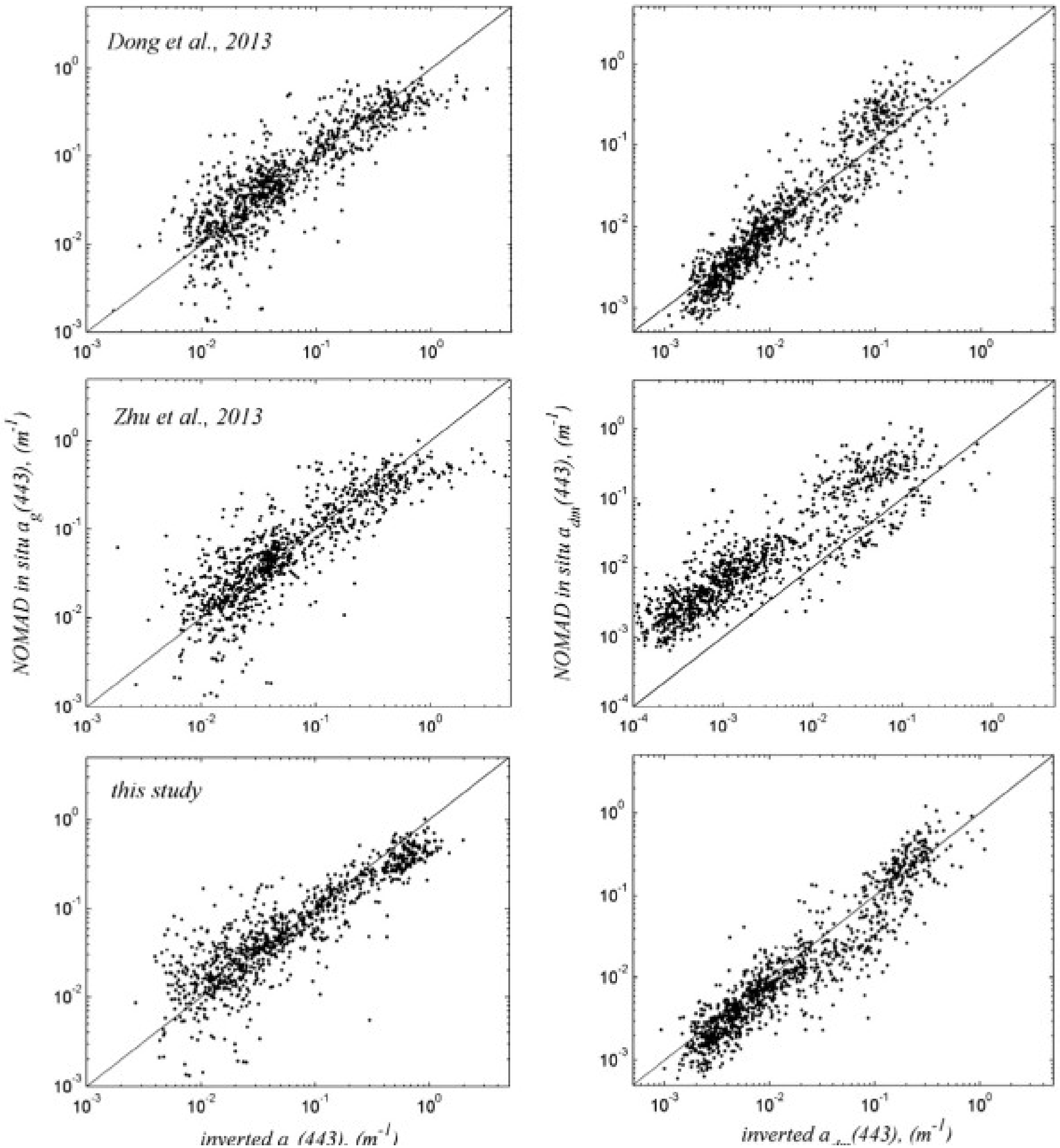


Pahlevan et al. (2020)

Applications

- Atmospheric correction of ocean color images
- Estimation of the chlorophyll-a concentration
- **Estimation of the Inherent Optical Properties of the seawater**
- Estimation of nutrients





Ioannou et al. (2013)

Applications

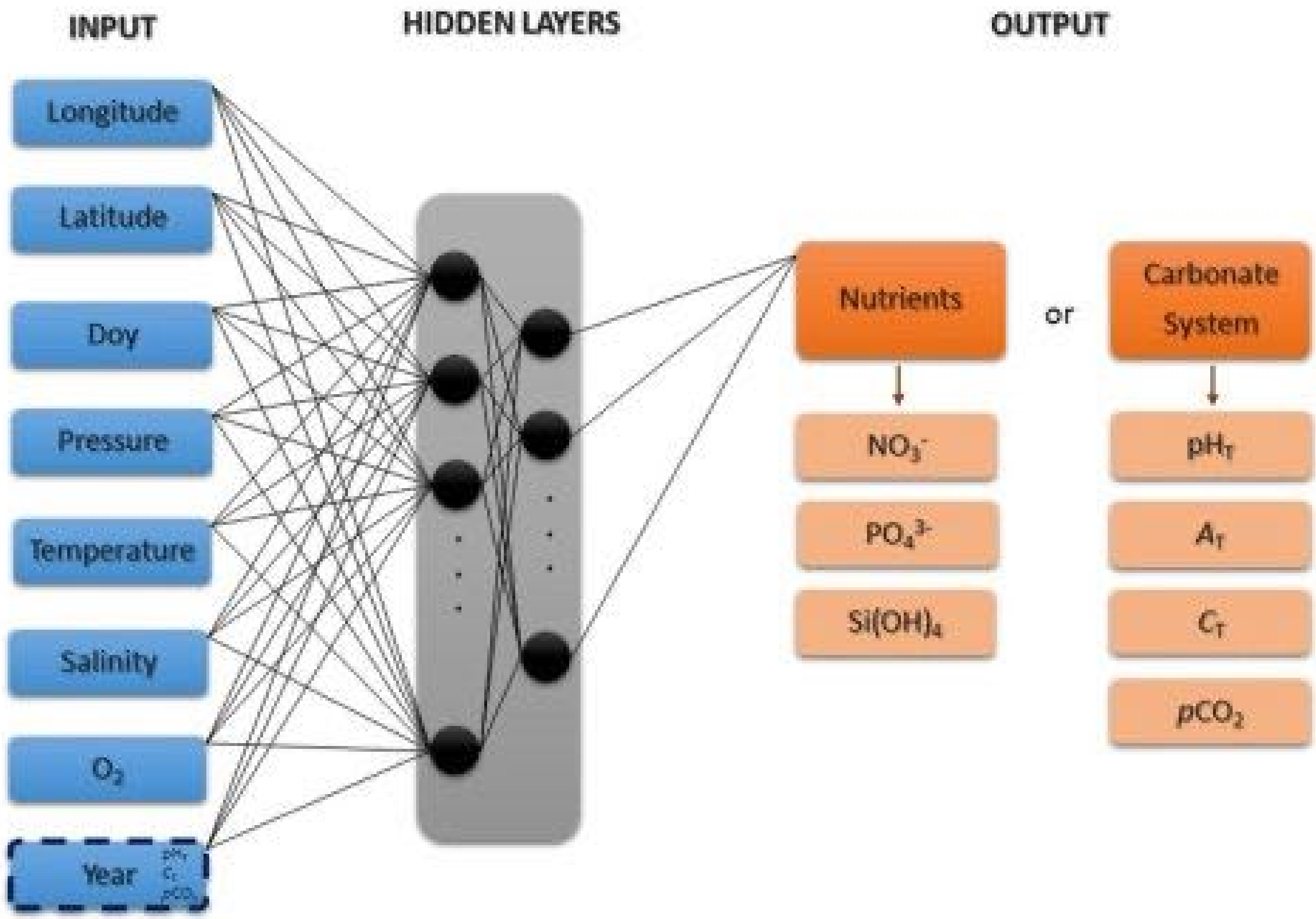
- Atmospheric correction of ocean color images
- Estimation of the chlorophyll-a concentration
- Estimation of the Inherent Optical Properties of the seawater
- **Estimation of nutrients**

CANYON method

- Using a MLP to estimate
  - Water-column (up to 8,000m depth) biogeochemically relevant variables in the global ocean
    - Concentrations of three nutrients (nitrate, phosphate and silicate) and four carbonate system parameters (total alkalinity, dissolved organic carbon, pH and pCO2)

CANYON method

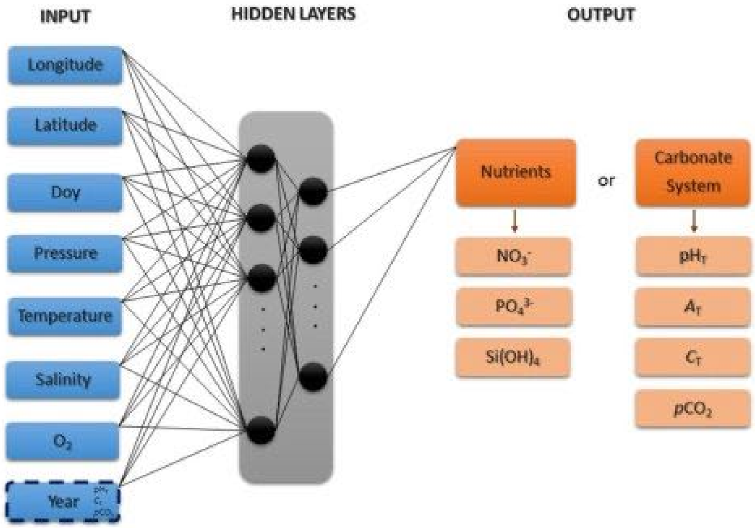
- Using a MLP to estimate
  - Water-column (up to 8,000m depth)

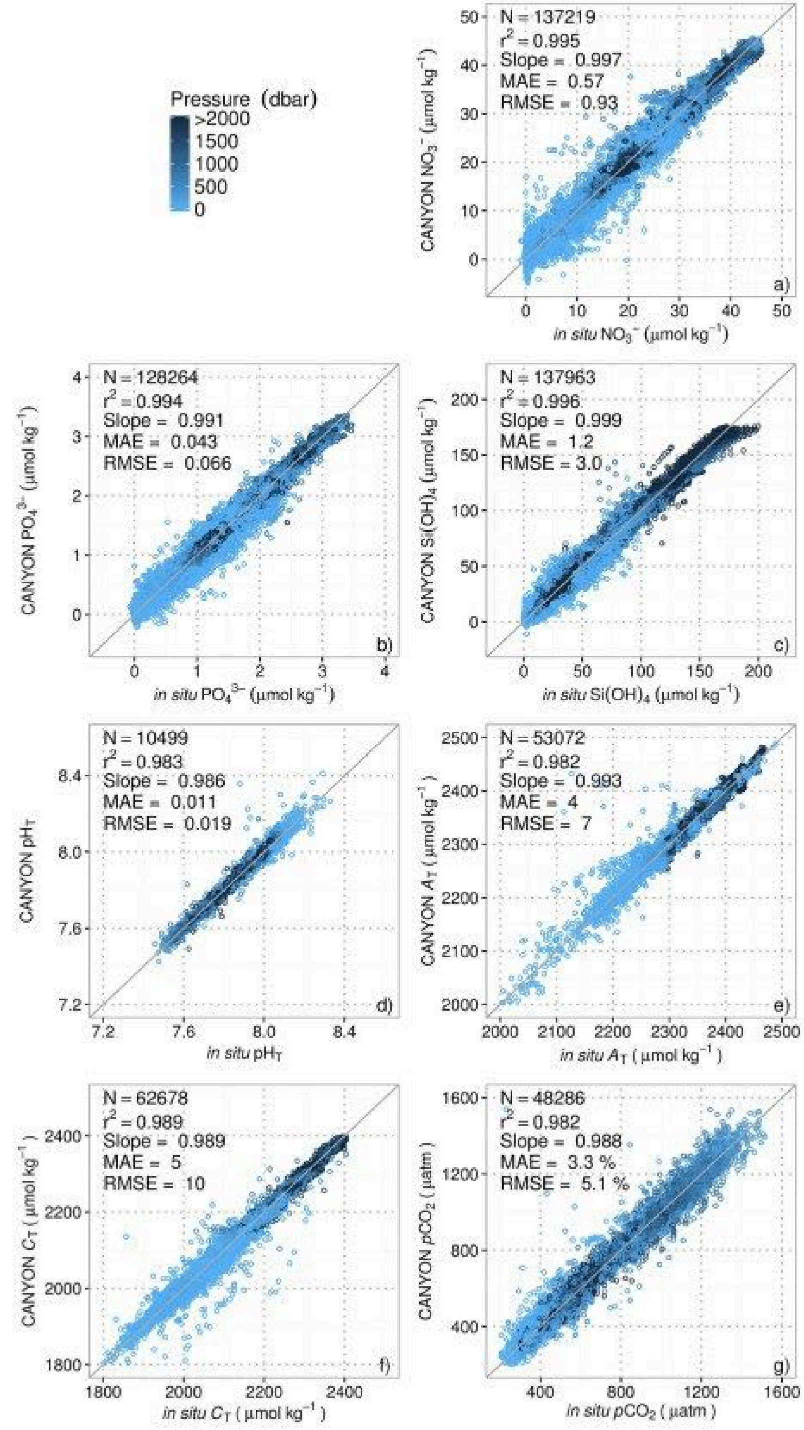


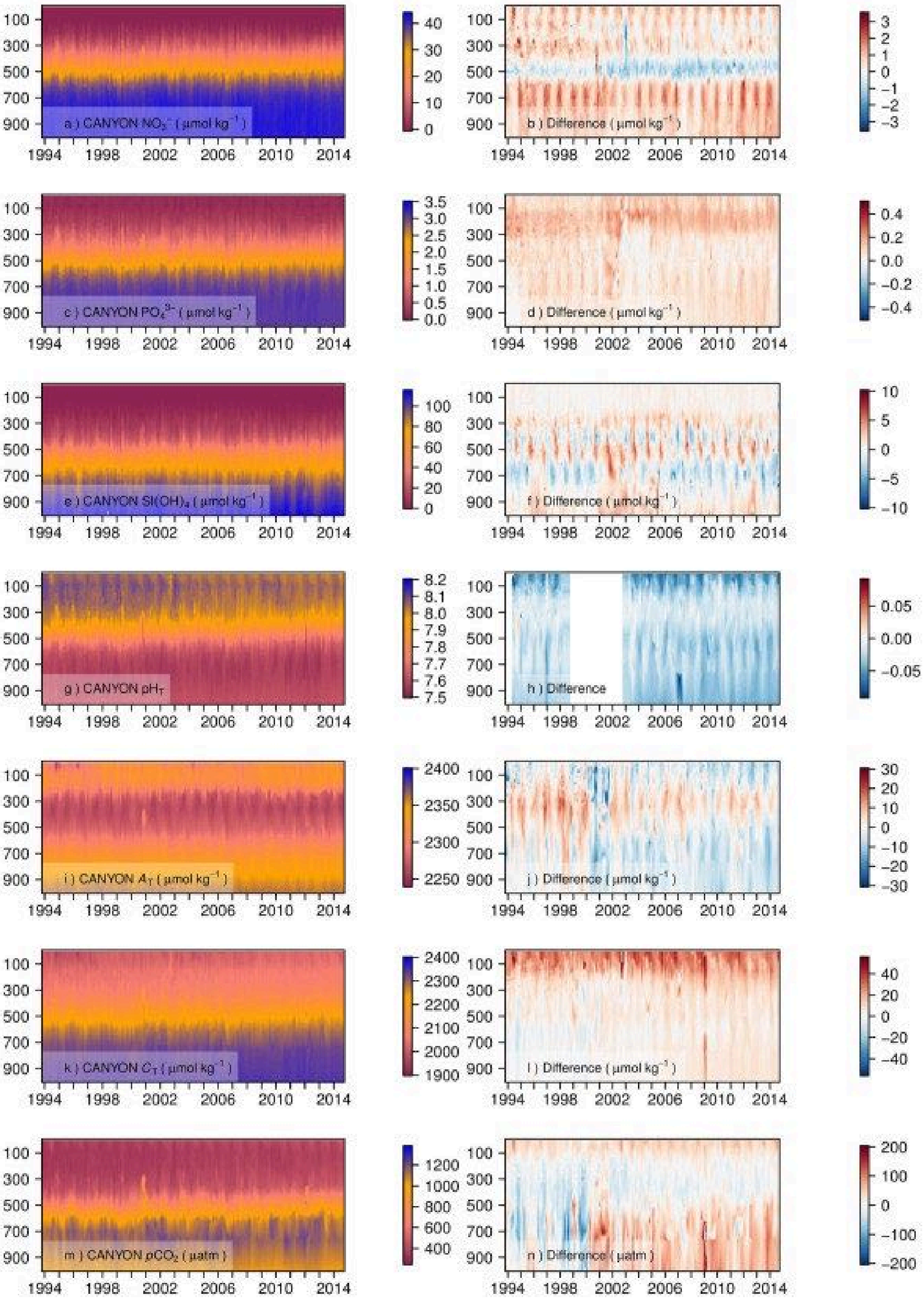
tem

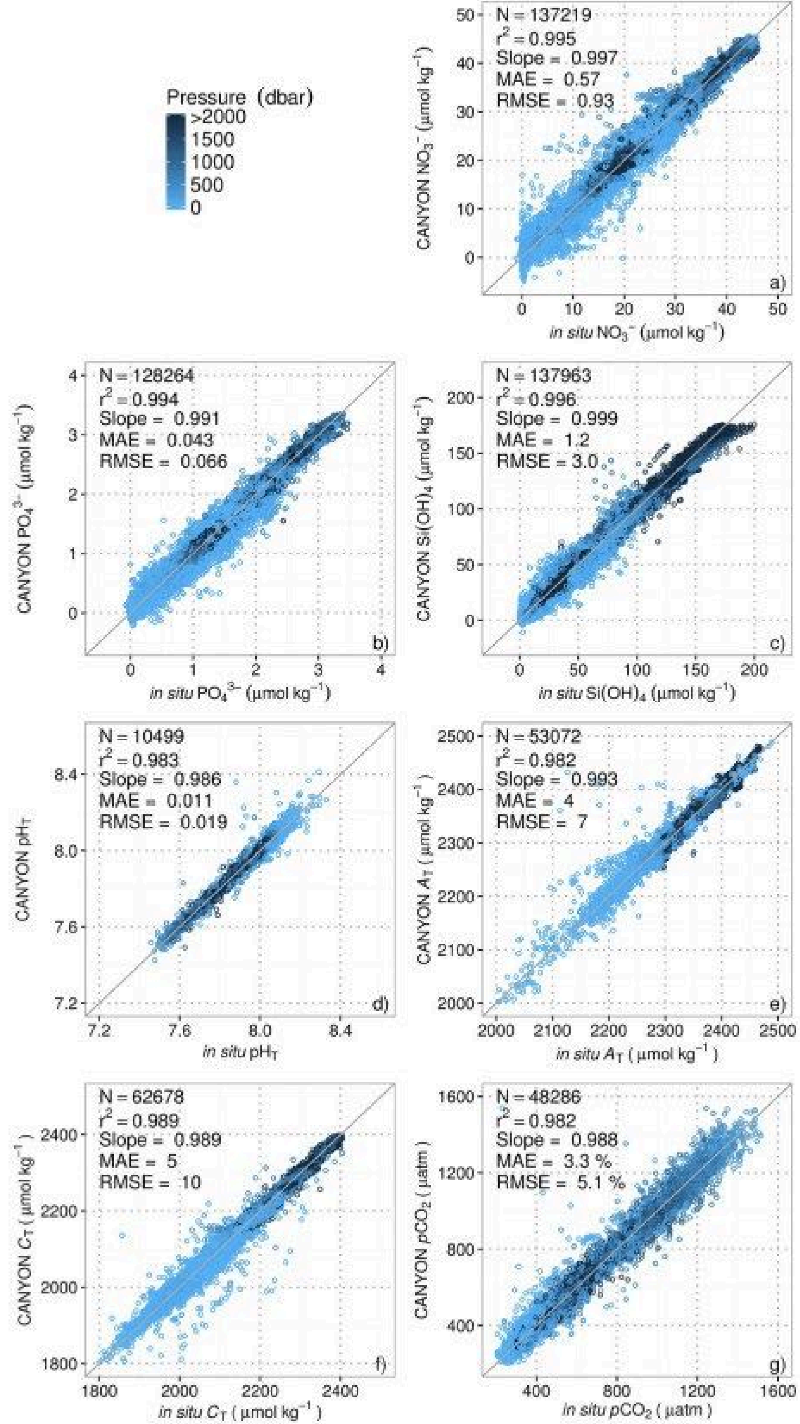
CANYON method

- Using a MLP to estimate
  - Water-column (up to 8,000m depth) biogeochemically relevant variables in the global ocean
    - Concentrations of three nutrients (nitrate, phosphate and silicate) and four carbonate system parameters (total alkalinity, dissolved organic carbon, pH and pCO2)
- Seven NN developed using GLODAPv2 database
  - Training using 80% randomly chosen
  - Remaining 20% used for validation


INPUT
HIDDEN LAYERS
OUTPUT
Longitude
Latitude
Doy
Pressure
Temperature
Salinity
O_2
Year
Nutrients
or
Carbonate System
NO_3^-
PO_4^{3-}
$Si(OH)_4$
pH_T
A_T
C_T
pCO_2


Pressure (dbar)
>2000
1500
1000
500
0
N = 137219
$r^2 = 0.995$
Slope = 0.997
MAE = 0.57
RMSE = 0.93
CANYON NO_3^- (µmol kg^{-1})
in situ NO_3^- (µmol kg^{-1})
a)
N = 128264
$r^2 = 0.994$
Slope = 0.991
MAE = 0.043
RMSE = 0.066
CANYON PO_4^{3-} (µmol kg^{-1})
in situ PO_4^{3-} (µmol kg^{-1})
b)
N = 137963
$r^2 = 0.996$
Slope = 0.999
MAE = 1.2
RMSE = 3.0
CANYON $Si(OH)_4$ (µmol kg^{-1})
in situ $Si(OH)_4$ (µmol kg^{-1})
c)
N = 10499
$r^2 = 0.983$
Slope = 0.986
MAE = 0.011
RMSE = 0.019
CANYON pH_T
in situ pH_T
d)
N = 53072
$r^2 = 0.982$
Slope = 0.993
MAE = 4
RMSE = 7
CANYON A_T (µmol kg^{-1})
in situ A_T (µmol kg^{-1})
e)
N = 62678
$r^2 = 0.989$
Slope = 0.989
MAE = 5
RMSE = 10
CANYON C_T (µmol kg^{-1})
in situ C_T (µmol kg^{-1})
f)
N = 48286
$r^2 = 0.982$
Slope = 0.988
MAE = 3.3 %
RMSE = 5.1 %
CANYON pCO_2 (µatm)
in situ pCO_2 (µatm)
g)


a) CANYON NO_3^- (µmol kg^{-1})
b) Difference (µmol kg^{-1})
c) CANYON PO_4^{3-} (µmol kg^{-1})
d) Difference (µmol kg^{-1})
e) CANYON $Si(OH)_4$ (µmol kg^{-1})
f) Difference (µmol kg^{-1})
g) CANYON pH_T
h) Difference
i) CANYON A_T (µmol kg^{-1})
j) Difference (µmol kg^{-1})
k) CANYON C_T (µmol kg^{-1})
l) Difference (µmol kg^{-1})
m) CANYON pCO_2 (µatm)
n) Difference (µatm)
1994 1998 2002 2006 2010 2014


Pressure (dbar)
>2000
1500
1000
500
0
N = 137219
r^2 = 0.995
Slope = 0.997
MAE = 0.57
RMSE = 0.93
CANYON NO_3^- (µmol kg^{-1})
in situ NO_3^- (µmol kg^{-1})
a)
N = 128264
r^2 = 0.994
Slope = 0.991
MAE = 0.043
RMSE = 0.066
CANYON PO_4^{3-} (µmol kg^{-1})
in situ PO_4^{3-} (µmol kg^{-1})
b)
N = 137963
r^2 = 0.996
Slope = 0.999
MAE = 1.2
RMSE = 3.0
CANYON $Si(OH)_4$ (µmol kg^{-1})
in situ $Si(OH)_4$ (µmol kg^{-1})
c)
N = 10499
r^2 = 0.983
Slope = 0.986
MAE = 0.011
RMSE = 0.019
CANYON pH_T
in situ pH_T
d)
N = 53072
r^2 = 0.982
Slope = 0.993
MAE = 4
RMSE = 7
CANYON A_T (µmol kg^{-1})
in situ A_T (µmol kg^{-1})
e)
N = 62678
r^2 = 0.989
Slope = 0.989
MAE = 5
RMSE = 10
CANYON C_T (µmol kg^{-1})
in situ C_T (µmol kg^{-1})
f)
N = 48286
r^2 = 0.982
Slope = 0.988
MAE = 3.3 %
RMSE = 5.1 %
CANYON pCO_2 (µatm)
in situ pCO_2 (µatm)
g)

Self-Organizing Maps (Unsupervised NN)

Unsupervised classification

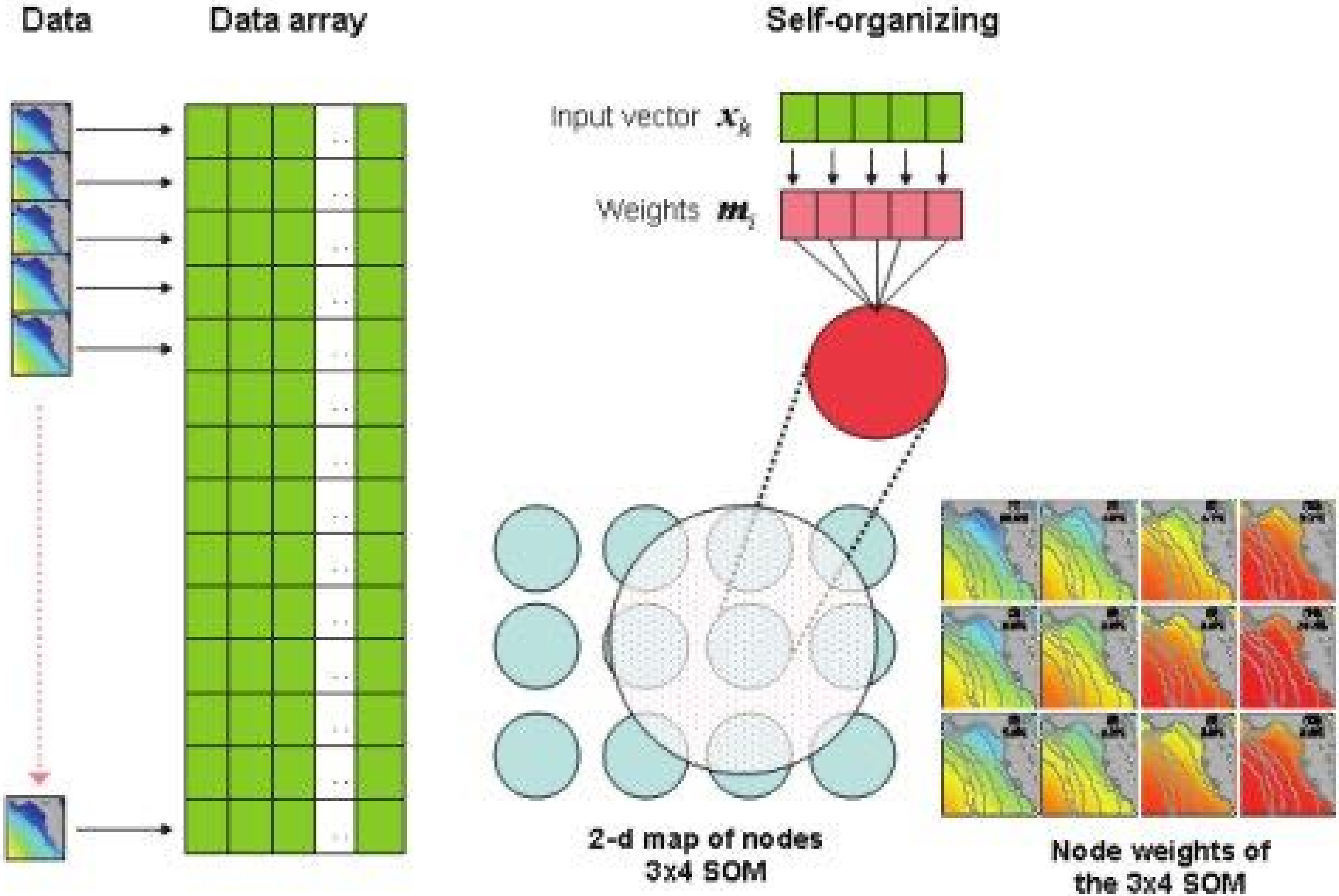
- Can solve non-linear problems of almost infinite complexity
- Robust in handling noisy data
- **Self-Organizing Maps (SOM):**
  - Produce a low-dimensional discretized representation of the input data
  - Fully automated method without any a priori knowledge on the data
  - Uses a neighborhood function to preserve the topological properties of the input space

Kohonen Self-Organizing Maps

- **Architecture:**
  - Kohonen maps consist of a two-dimensional array of neurons, fully connected, with no lateral connections, arranged on a squared or hexagonal lattice
- **Learning algorithm:**
  - follows the winner-take-all strategy
  - forces close neurons to fire for similar inputs (Self-Organizing Maps)
- **Properties:**
  - The topology of the input space is preserved

Self organizing maps

- The purpose of SOM is to map a multidimensional input space onto a topology preserving map of neurons
  - Preserve a topological so that neighboring neurons respond to «similar »input patterns
  - The topological structure is often a 2 or 3 dimensional space
  - the distance and proximity relationship (i.e., topology) are preserved as much as possible
- Similar to specific clustering: cluster centers tend to lie in a lowdimensional manifold in the feature space



Liu and Weisberg (2011)

Self organizing maps

- The SOM models are associated with the nodes of a regular, usually two-dimensional grid
- SOM constructs the models such that:
  - More similar models will be associated with nodes that are closer in the grid, whereas less similar models will be situated gradually farther away in the grid.
- Central idea of SOM
  - Every input data item shall select the model that matches best with the input item, and this model, as well as a subset of its spatial neighbors in the grid, shall be modified for better matching.

Self organizing maps

- **First step of the classification:**
  - To summarize the information contained in the training dataset by producing a set of reference vectors
    - Compression of the information embedded in the dataset
    - Each neuron is associated with a particular reference vector and thus corresponds to a group of specific sub-dataset
  - Neurons of the topological map are connected and determine the topological (neighborhood) relationship among the different neurons (groups)
    - Close neurons on the map represent similar subsets of data (classes presenting similarities)

Self organizing maps

- **Second step of the classification:**
  - Making link between the reference vectors and the parameters of interest
  - Supervised (experts) phase

Self organizing maps

- The activation of the neuron is spread in its direct neighborhood →neighbors become sensitive to the same input patterns
- The size of the neighborhood is initially large but reduce over time → Specialization of the network

Self organizing maps

- **Two quantitative measures of mapping quality are commonly used**:
  - *Quantization Error (QE)*: Average distance between each data point
  - *Topographic Error (TE)*: Fraction of data points for which the first *Best Matching Unit* and the second *Best Matching Unit* are not neighbouring units
  - Smaller QE and TE values indicate better mapping quality → Used to choose the appropriate number of neurons in the SOM

Self organizing maps

- Only the batch-learning version of the SOM is recommendable for practical applications, because it does not involve any learning-rate parameter, and its convergence is an order of magnitude faster and safer

Applications

Oceanographic data	Regions	References
Satellite ocean color, Chlorophyll	Pacific	Ainsworth (1999), Ainsworth & Jones (1999)
	Southeast Atlantic	Yacoub et al. (2001)
	Southwest Atlantic	Saraceno et al. (2006)
	North Atlantic	Niang et al. (2003), Telszewski et al. (2009)
In situ Chlorophyll, absorption spectra of phytoplankton, plankton taxa, ecological variables, microbiological and geochemical variables, pCO2	Southern North Sea	Kropp & Klenke (1997)
	Southeast Atlantic	Silulwane et al. (2001), Richardson et al. (2002)
	Europe	Barreto & Perez-Uribe (2007ab), Alvarez-Guerra et al. (2008), Aymerich et al. (2009), Skwarzec et al. (2009), Žibret & Šajn (2010)
	Lagoon of Venice	Bandelj et al. (2008)
	Northern Adriatic Sea	Solidoro et al. (2007, 2009)
	Antarctic	Lee et al. (2003)
	World oceans	Chazottes et al. (2006, 2007)
	Indian Ocean	Astel et al. (2008)
Satellite measured sea surface temperature, ENSO indices	Pacific	Ainsworth (1999), Ainsworth & Jones (1999)
	Tropical Pacific	Leloup et al. (2007, 2008)
	Southeast Atlantic	Richardson et al. (2003)
	West Florida Shelf	Liu et al. (2006b)
	North Atlantic	Telszewski et al. (2009)
	Indian Ocean	Tozuka et al. (2008), Morioka et al. (2010), Iskandar (2010)
Satellite measured sea surface height	Southeast Atlantic	Hardman-Mountford et al. (2003)
	South China Sea	Liu et al. (2008)
	Indian Ocean	Iskandar (2009)
Ocean currents from in situ observations and numerical models	West Florida Shelf	Liu & Weisberg (2005, 2007), Liu et al. (2006a, 2007)
	Columbia River plume	Liu et al. (2009)
	New York Harbor	Cheng & Wilson (2006)
	New York Bight	Mau et al. (2007)
	Indian Ocean	Iskandar et al. (2008)
	Kerama Gap	Jin et al. (2010)
Surface winds	Southeast Atlantic	Richardson et al. (2003), Risien et al. (2004)
Sea-floor roughness	South Atlantic	Chakraborty et al. (2003)
Salinity	Columbia River plume	Liu et al. (2009)
Oil spill	Galician coast, Europe	Corchado et al. (2008), Mata et al. (2009), Borges et al. (2010)
Maritime data	Europe	Lobo (2009)

Table 2. SOM applications in oceanography

Liu and Weisberg (2011)

Applications

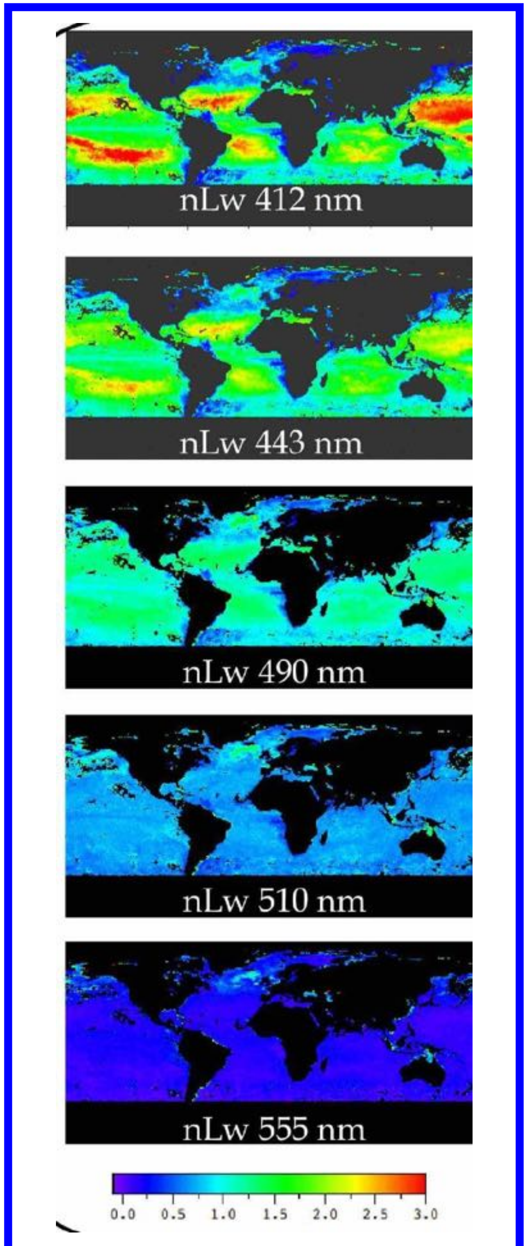
- Phytoplantkon Functional Types

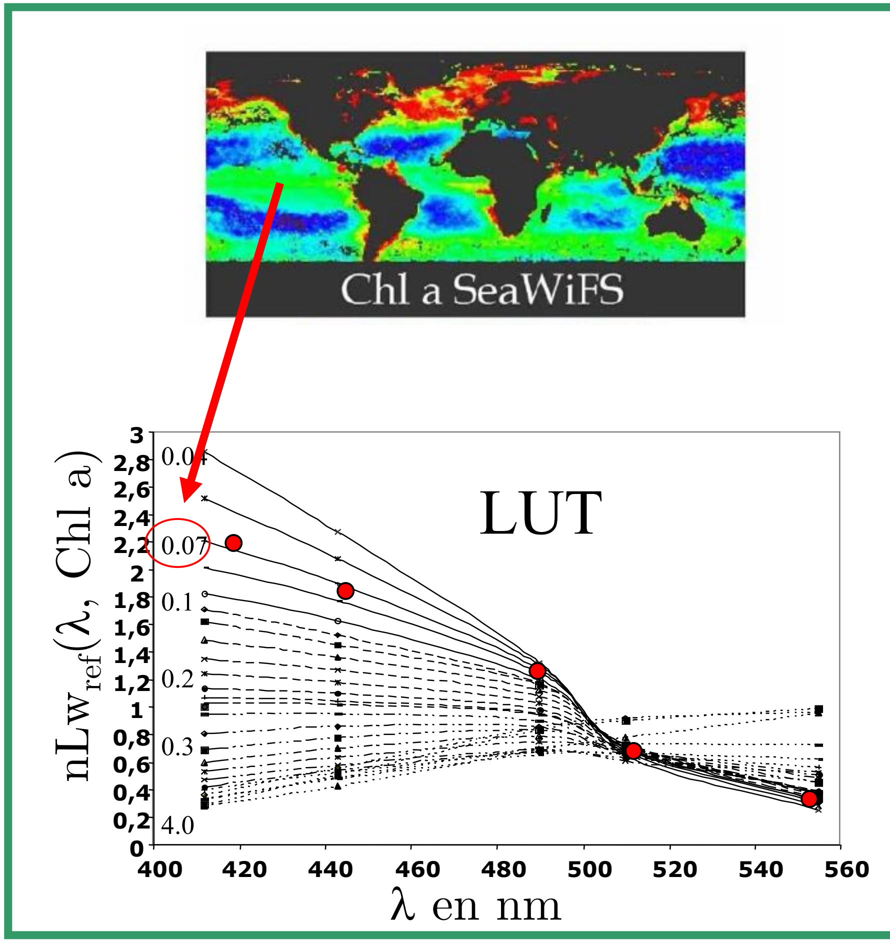
PHYSAT method

- The objective was to propose a global method in order to detect a large variety of phytoplankton functional group (biogechemical function) when they are dominant.

⇒ This method doesn't rely on IOP or chlorophyll *a* assumption or specific regional algorithms.

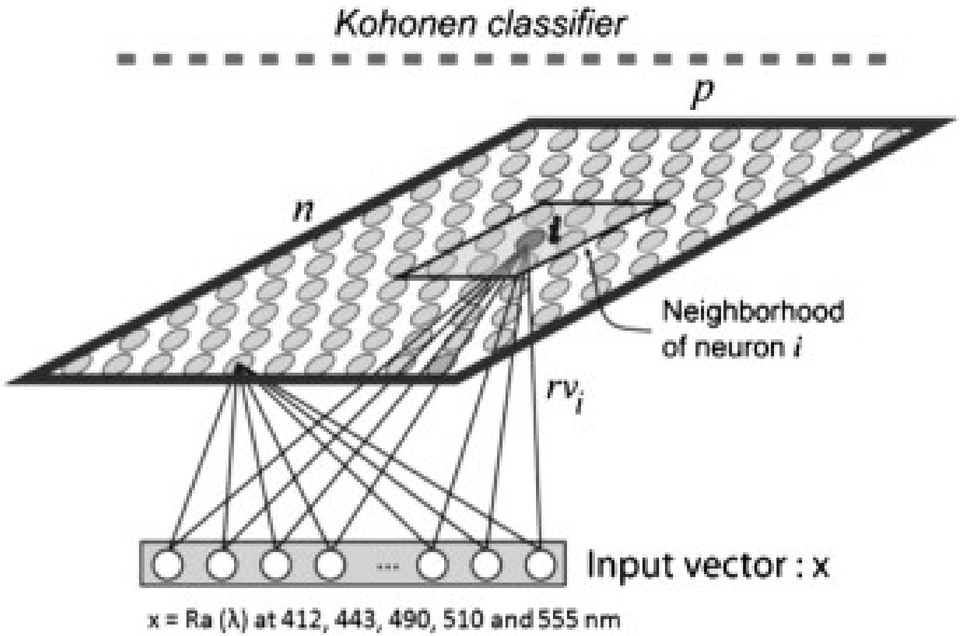
Removed the first order Chl a effect on the signal





$$nLw^*(\lambda) = nLw(\lambda) / nLw_{ref}(\lambda,\ Chl\ a)$$

Ben Mustapha et al. (2014)



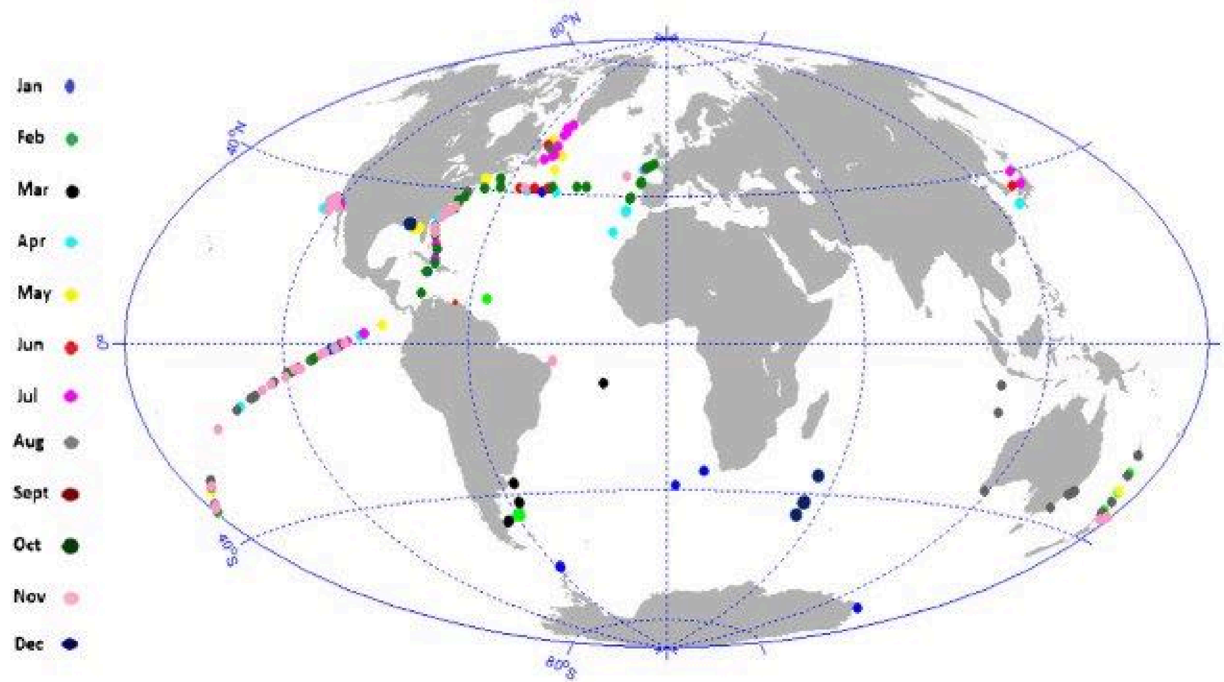
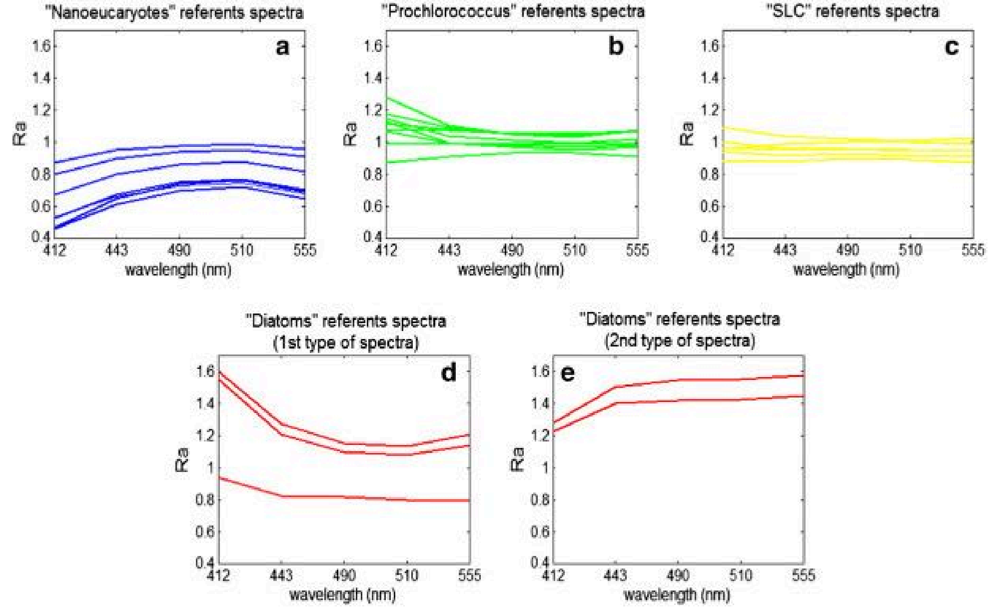
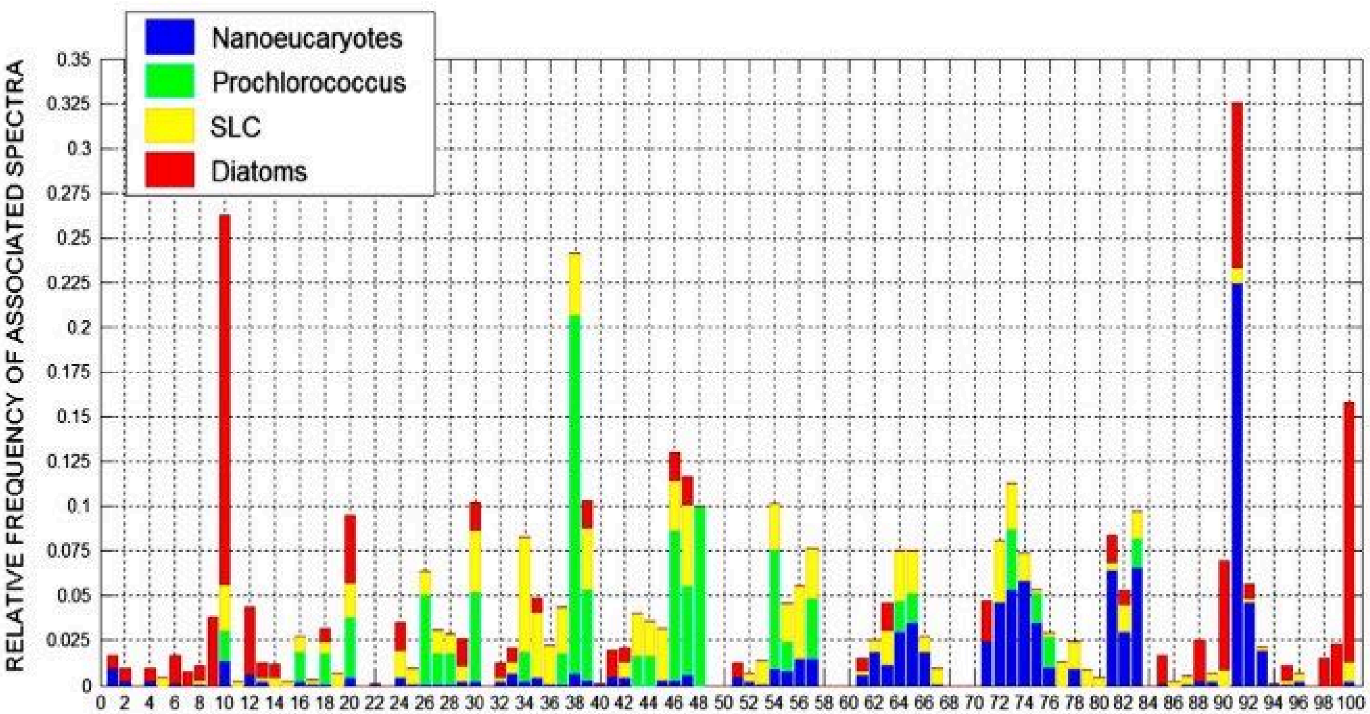


Fig. 1. Geographic distribution of the 1068 matchups, each month has been colored differently in order to show the seasonal variability covered by the in situ data.



"Nanoeucaryotes" referents spectra
"Prochlorococcus" referents spectra
"SLC" referents spectra
"Diatoms" referents spectra (1st type of spectra)
"Diatoms" referents spectra (2nd type of spectra)
a
b
c
d
e
Ra
wavelength (nm)
412
443
490
510
555



Nanoeucaryotes
Prochlorococcus
SLC
Diatoms
RELATIVE FREQUENCY OF ASSOCIATED SPECTRA

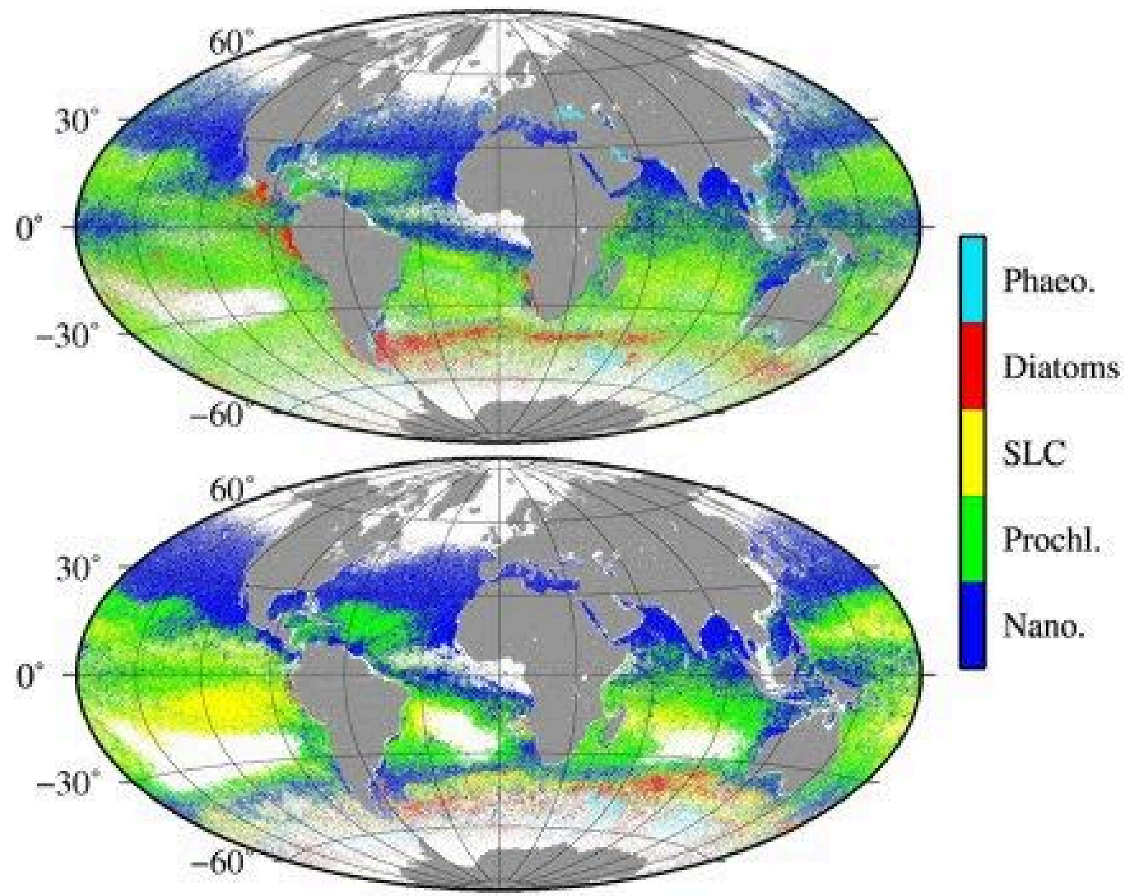
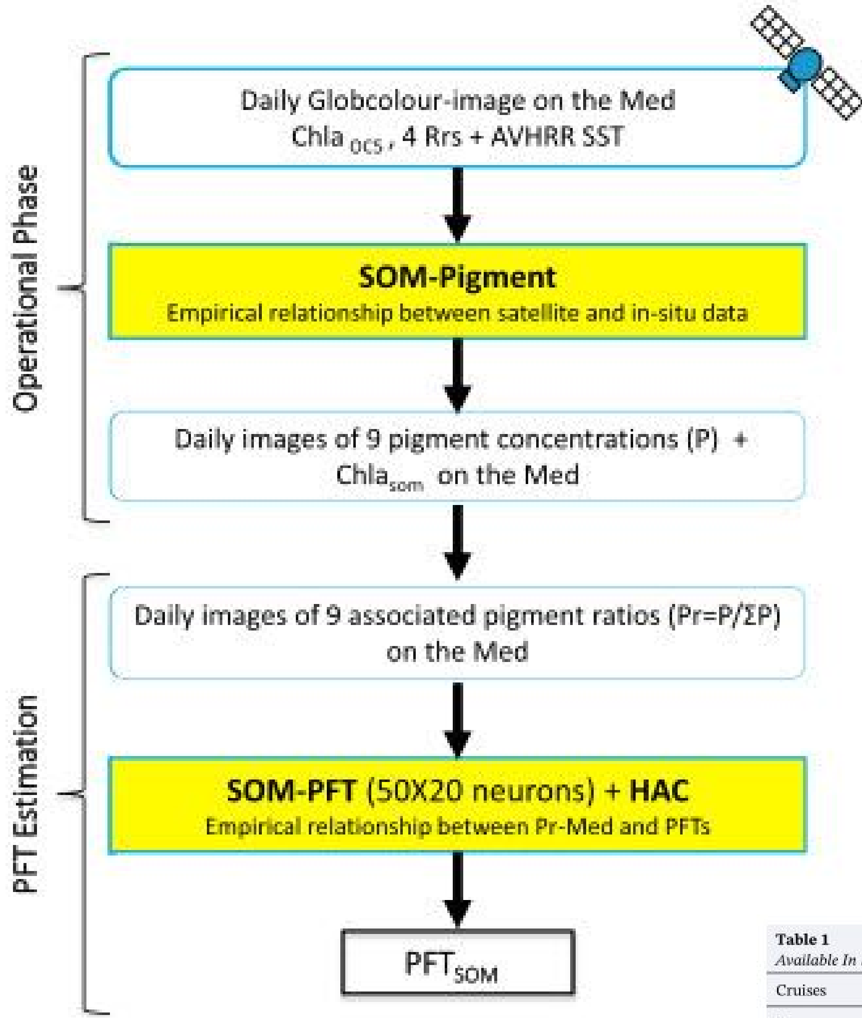


Fig. 11. Global climatology of most frequently dominant phytoplankton groups over the 1997–2010 SeaWiFS image archive made by PHYSAT-SOM (top) and PHYSAT-v2008 (bottom) for January month.

PFT in Mediterranean Sea



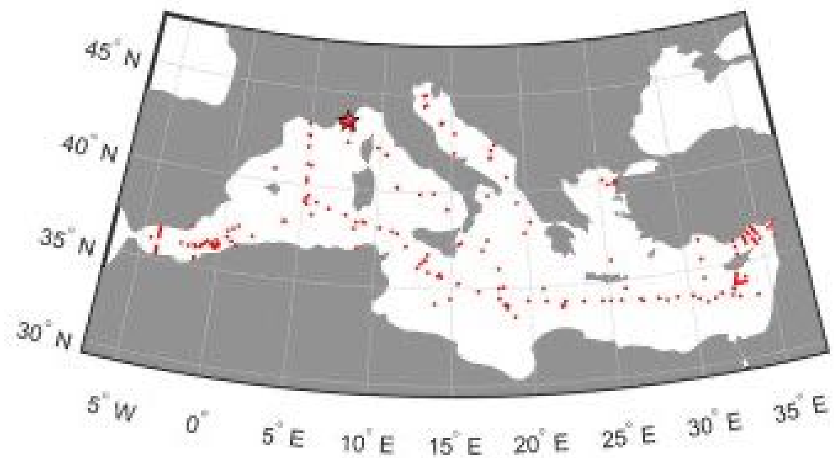


Figure 1. Localities of the HPLC samples in the Mediterranean Sea (Boussole station is represented with a star).

Table 1
Available In Situ HPLC Inventory in the Mediterranean Sea

Cruises	Location	Period	N	%	Source
Prosope	Western basin	9/4/1999 to 4/10/1999	59	3	a
SODYFT	Ligurian Sea	02/25/2002 to 12/19/2005	160	9	
SESAME	Mediterranean Sea	02/16/2008 to 10/19/2008	261	15	
BOUM_bot	Mediterranean Sea	06/21/2008 to 07/18/2008	64	4	b
Tara_oceans	Mediterranean Sea	9/20/2009 to 10/27/2013	115	6	
BOUSSOLE	Ligurian Sea	07/22/2001 to 11/10/2016	1,113	63	c
Total			1,772		

Note. Dates are formatted as MM/DD/YYYY.
[a]https://doi.org/10.5194/essd-5-109-2013 [b]https://seabass.gsfc.nasa.gov/ [c]http://www.obs-vlfr.fr/Boussole/html/boussole_data/collected.php

El Hourani et al. (2019)

PFT in Mediterranean Sea

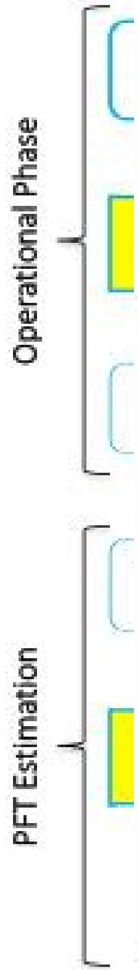


Table 2
Statistical Results of the Cross-Validation Procedure Performed by the SOM-Pigments Using the Med HPLC Data Set

Pigment	R^2	RMSE (mg/m^3)	*N* Obs
$Chla_{SOM}$	0.81	0.21	1,113
DVChla	0.72	0.007	663
Chlb	0.78	0.015	858
DVChlb	0.74	0.0005	79
19HF	0.66	0.023	1,030
19BF	0.80	0.006	1,096
Fuco	0.85	0.044	1,133
Perid	0.80	0.006	890
Allo	0.62	0.014	579
Zea	0.82	0.008	1,241

Note. RMSE = root-mean-square error; SOM = self-organizing map.

					Source
					a
					b
Tara_oceans	Mediterranean Sea	9/20/2009 to 10/27/2013	115	6	c
BOUSSOLE	Ligurian Sea	07/22/2001 to 11/10/2016	1,113	63	
Total			1,772		

Note. Dates are formatted as MM/DD/YYYY.
[a]https://doi.org/10.5194/essd-5-109-2013 [b]https://seabass.gsfc.nasa.gov/ [c]http://www.obs-vlfr.fr/Boussole/html/boussole_data/collected.php

El Hourani et al. (2019)

PFT in Mediterranean Sea

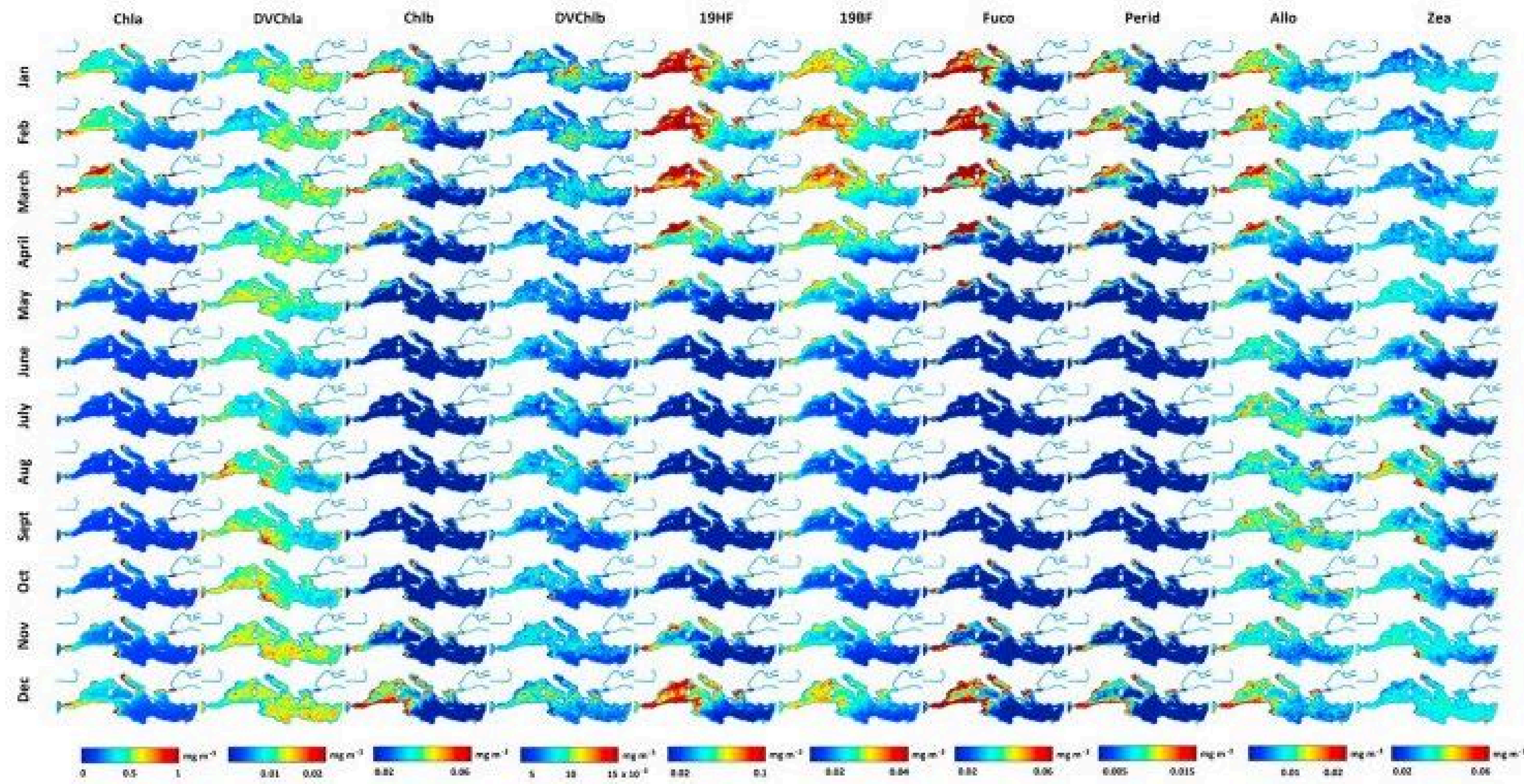


Figure 4. Monthly climatology images for each estimated pigment concentrations by self-organizing map-pigments along with the Chla OC5.

PFT in Mediterranean Sea

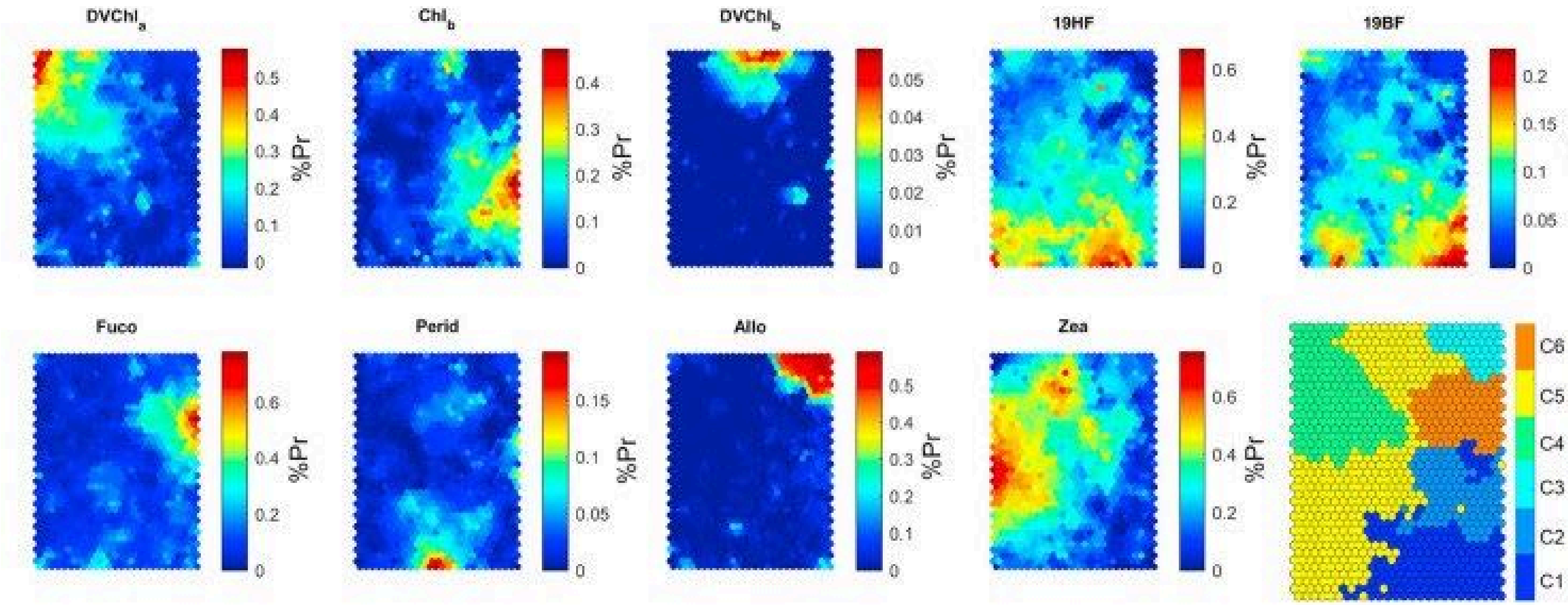


Figure 5. Discrete representation of each Pr on the self-organizing map-phytoplankton functional type and the hierarchical clustering in six classes (bottom right panel).

El Hourani et al. (2019)

Thank you