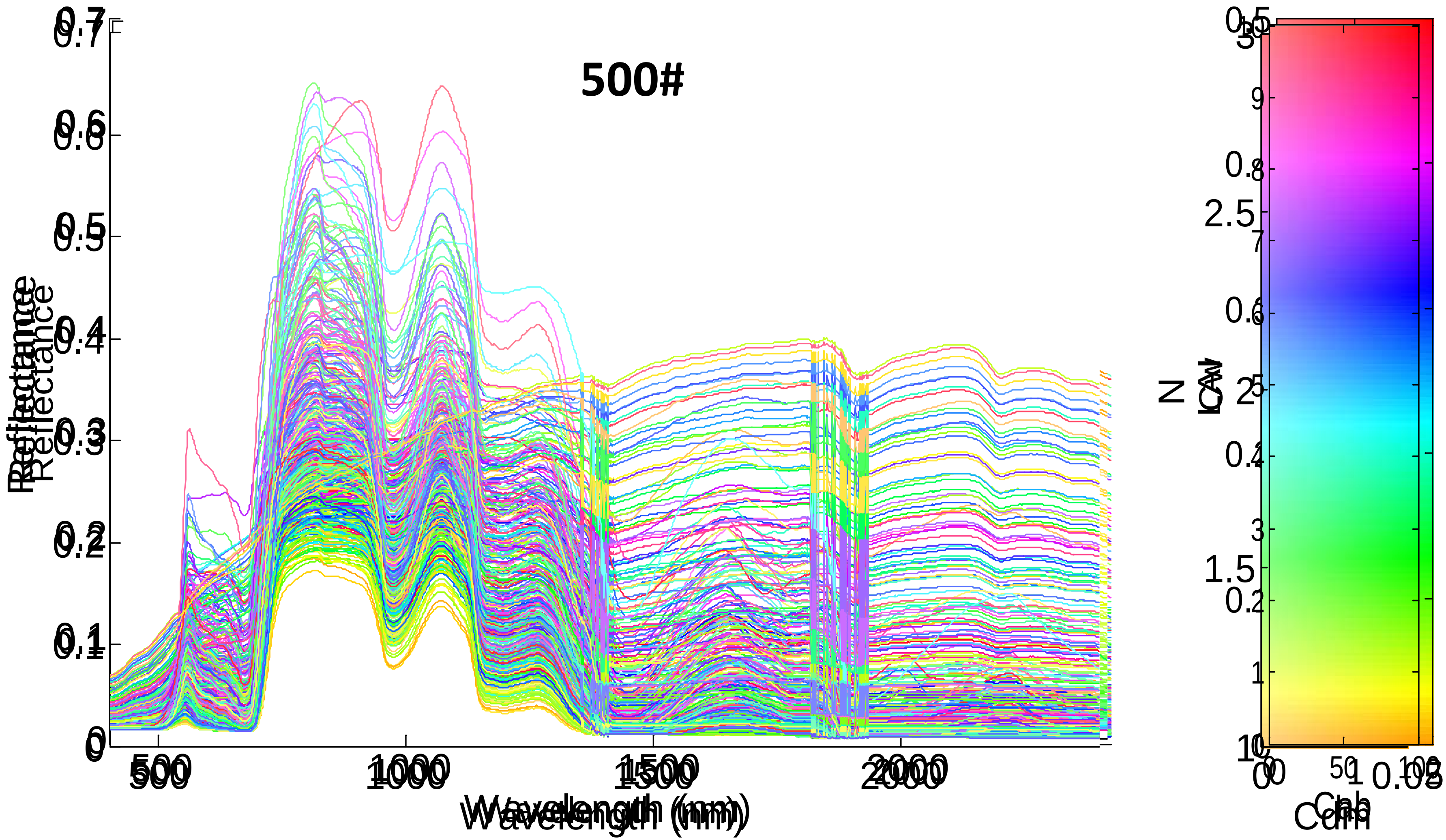
# Emulation of radiative transfer models (RTMs):

## new opportunities for spectroscopy data processing

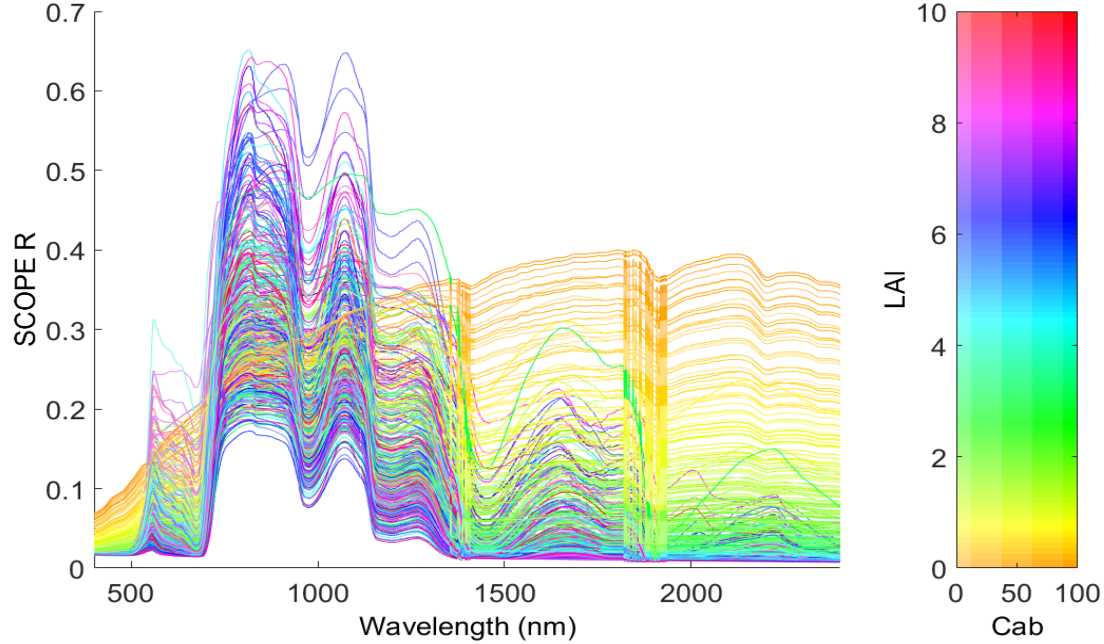
*Jochem Verrelst, Juan Pablo Rivera, Jordi Muñoz-Marí, Neus Sabater, Jorge Vicent, Jose Moreno and Gustau Camps-Valls*
Image Processing Laboratory, Univ. of Valencia (Spain)

EARSeL Imaging Spectroscopy Workshop
19 April2017


500#
Reflectance
Wavelength (nm)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
500
1000
1500
2000
N
Cab
Cm

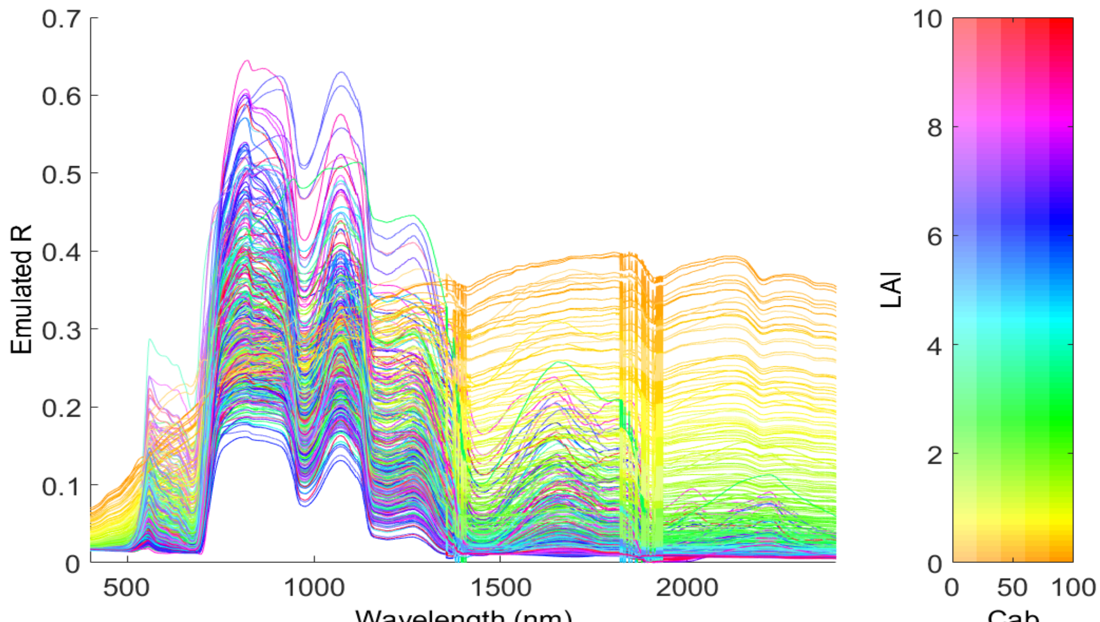
# Any difference? Which model would you choose?

**37 min**




# SCOPE

**9 s**



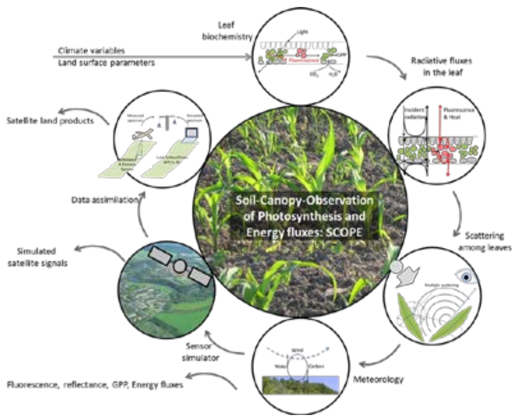

# Emulator

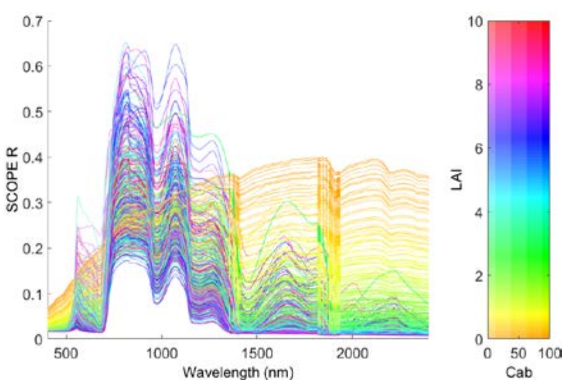
**(emulated SCOPE)**

# BACKGROUND

## Advanced RTMs: generation of a large LUT (>1000#)

### SCOPE

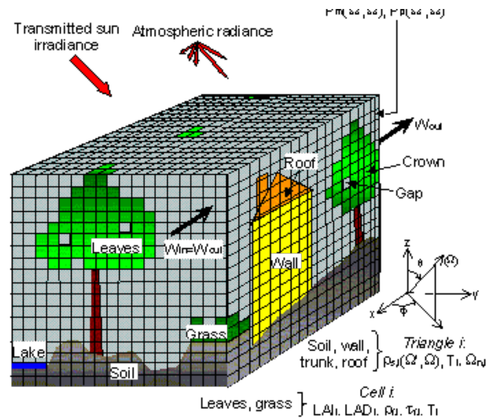


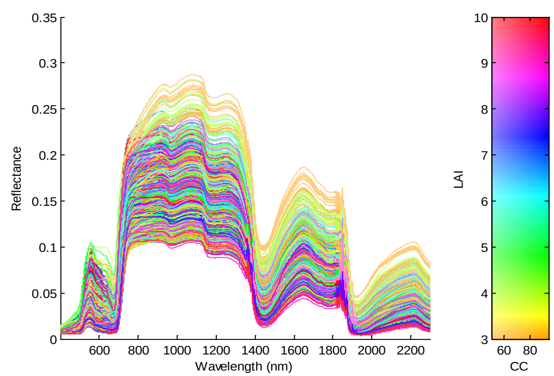





# Hours

### DART

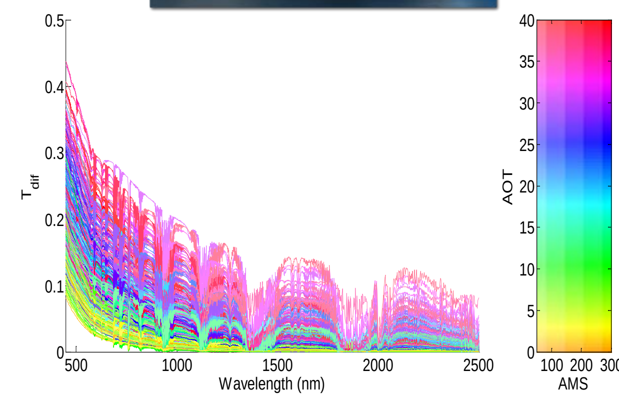




# days

### MODTRAN



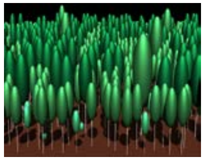
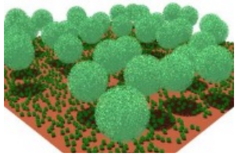
# >days

# Advanced RTMs: *more realistic but slow*

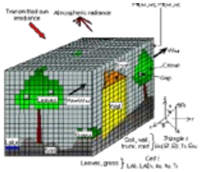
- **RTMs** are widely used in RS science, e.g. for development of new missions and retrieval (**inversion**).
- When choosing an **RTM**, **a trade-of between applicability and realism** has to be made: **simpler models** are easier to apply but **less realistic**, while **advanced models are more realistic but require a large amount of variables to be configured**.

<u>**Examples of advanced RTMs:**</u>

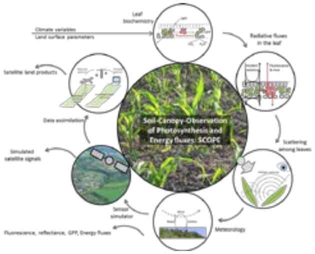
- **Ray tracing models** (e.g. FLIGHT, RAYTRAN)

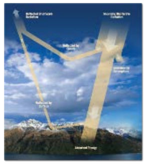

- **Voxel models**: DART



- **Soil-Vegetation-Atmosphere-Transfer (SVAT) models:** e.g. SCOPE



- **Atmospheric transfer models:** e.g. MODTRAN




- **Main drawback of advanced RTMs involves their long processing time**: ***the more advanced, the longer it takes to generate output.***
- Long processing time makes that **advanced RTMs are of little use for operational tasks**, e.g., pixel-by-pixel inversion schemes.

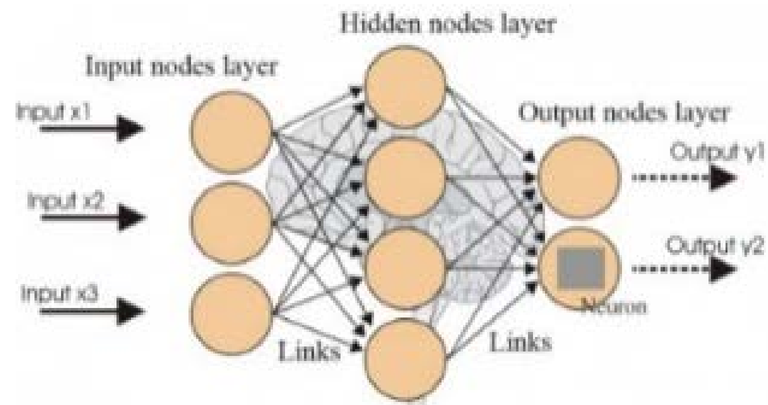
# Emulation of RTMs

**<u>Emulators</u> are statistical models that are able to approximate the processing of a physical model (e.g. RTM) - <u>at a fraction of the computational cost</u>:**

*making a statistical model of a physical model*

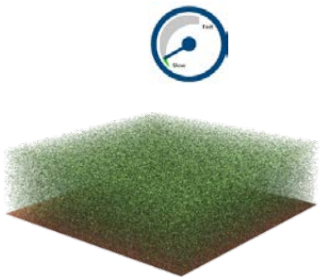
**RTM**

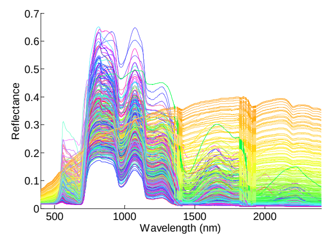
**Machine learning**

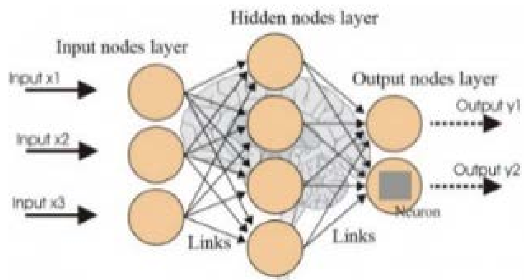



# Regression vs. Emulation:

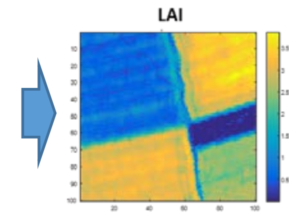
## Classical use of machine learning in optical RS:





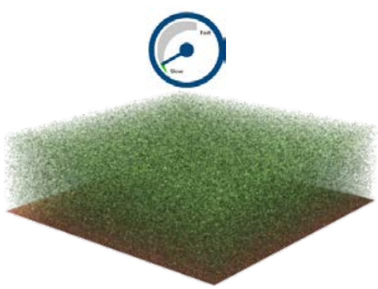

Variable (e.g. LAI)

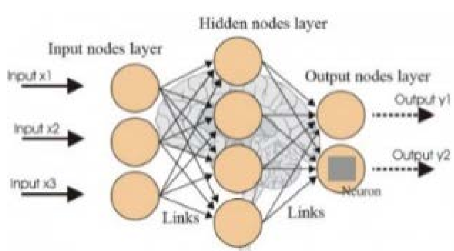


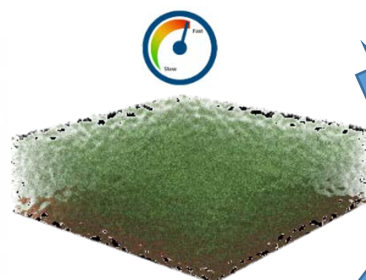

## Statistical method:

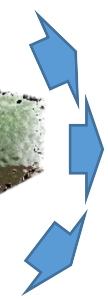
- Variable/data-driven, black box, 1 output, portability is questionable

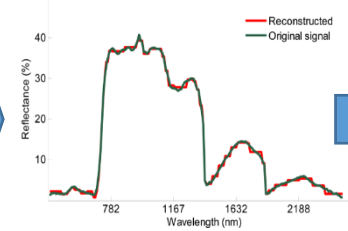
## Emulation in optical RS:

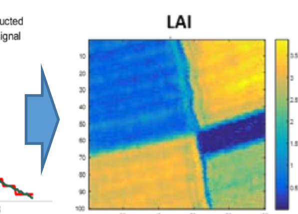








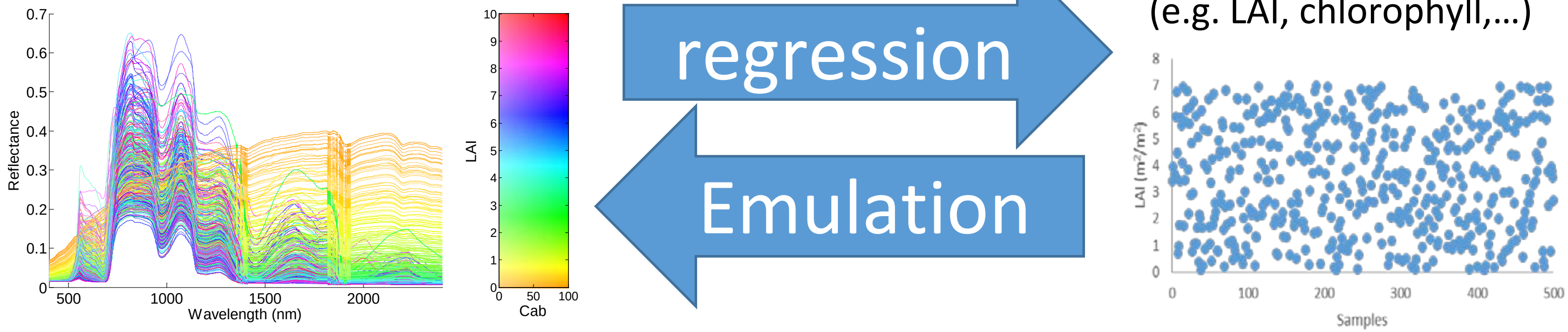




## Replace RTM:

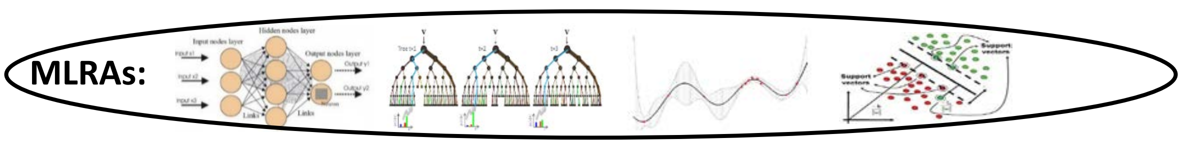
- Multiple applications, e.g. inversion
  - ✓ Radiometric method: Spectral fitting
  - ✓ Portable: generally applicable

# Emulation in practice:

**Spectra** | **Variables** (e.g. LAI, chlorophyll,...)




- In principle any **nonlinear, adaptive machine learning regression algorithm (MLRAs) can serve as emulator**.




- *However*, **to emulate RTM spectral output,** the MLRA should have the capability to reconstruct **multiple outputs**, i.e. the complete spectrum. *Only Few MLRAs possess multi-output capability (e.g. NN).*

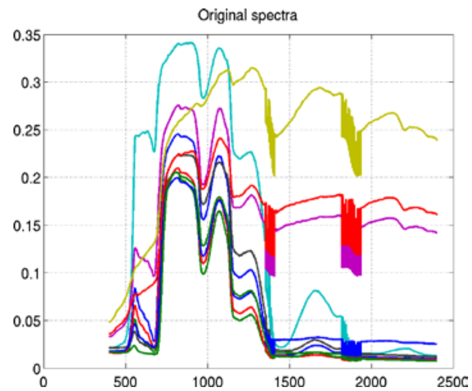
- **Multi-output resolved with dimensionality reduction techniques** (e.g. PCA).

**"spectral redundancy" is a blessing**

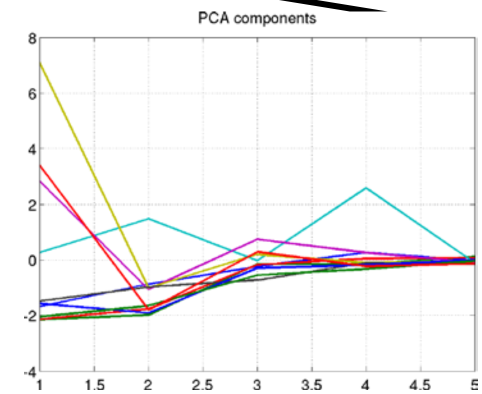
# Processing steps

Input (variables + spectra) → Splitting into training/validation → PCA on spectra → MLRA training looping over components → Prediction of components → Reconstruction of spectra → Validation → Emulator

**PCA on spectra**




$$Sc = U \cdot X$$



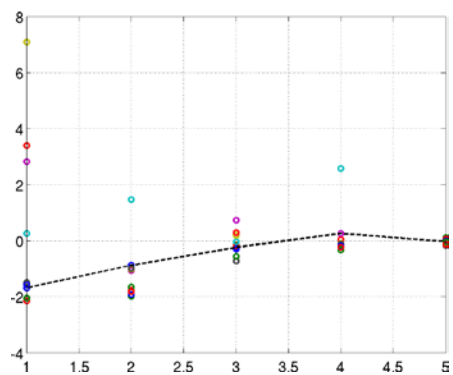

**MLRA training looping over components**

$$W = (Y + \lambda I)^{-1} \cdot Sc$$

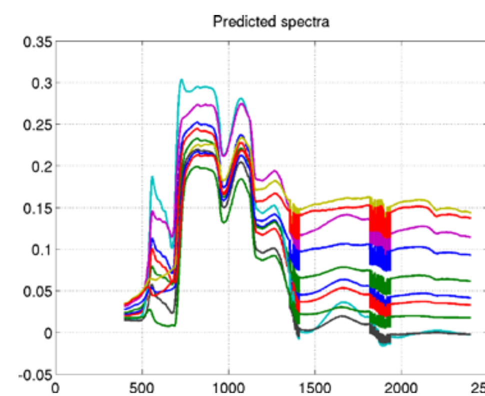
**Prediction of components**

$$Sp = Sc \cdot W$$

**Reconstruction of spectra**



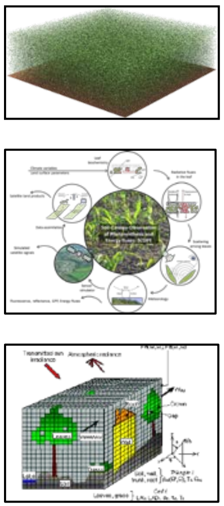
$$Xr = U^{T} \cdot Sp$$

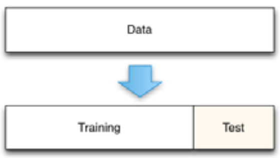


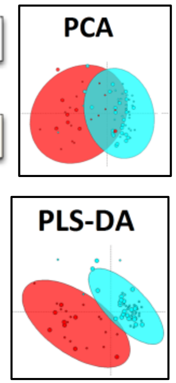

Validation → Emulator

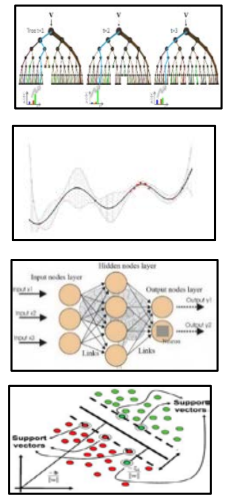
# Emulator toolbox

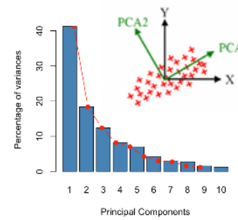
**With ARTMO's emulation processing chain any RTM can be converted into an emulator.**



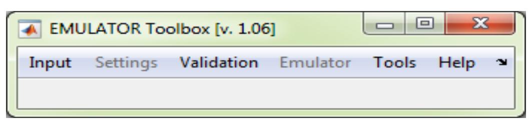





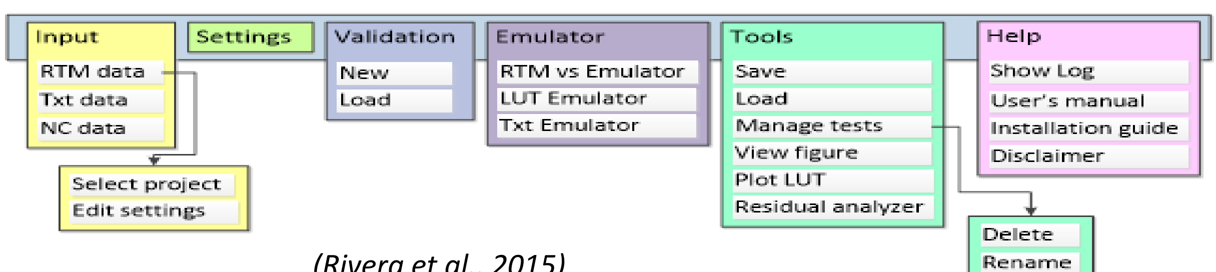

$$\mathrm{Xr} = \mathrm{U}^{\mathrm{T}} \cdot \mathrm{Sp}$$

$$\mathrm{RMSE} = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}}$$

$$\mathrm{NRMSE} = \frac{1}{Y_{\max} - Y_{min}} \sqrt{\frac{\sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2}{N}},$$




**Input** → **Settings** → **Validation** → **Emulator**

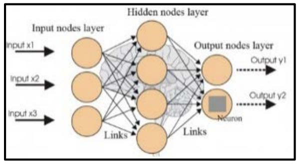


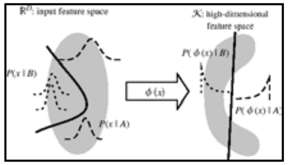

*(Rivera et al., 2015)*

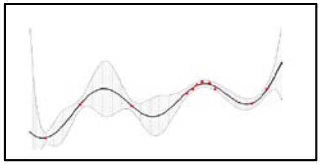
# *Emulators great idea… what about accuracy?*

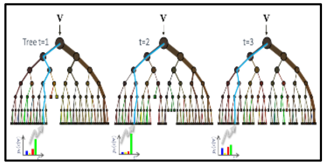
## Various open questions:

1) *Role of machine learning regression algorithm?*

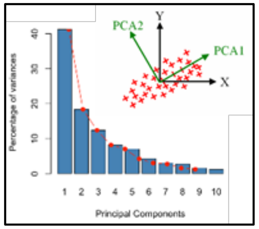


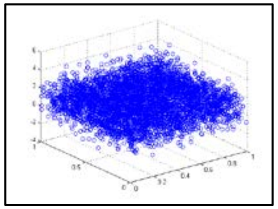





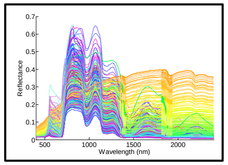
2) *Role of dimensionality reduction (DR) method?*

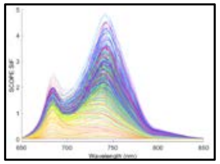


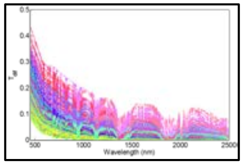

3) *Role of LUT size training?*



4) *Role of data type?*

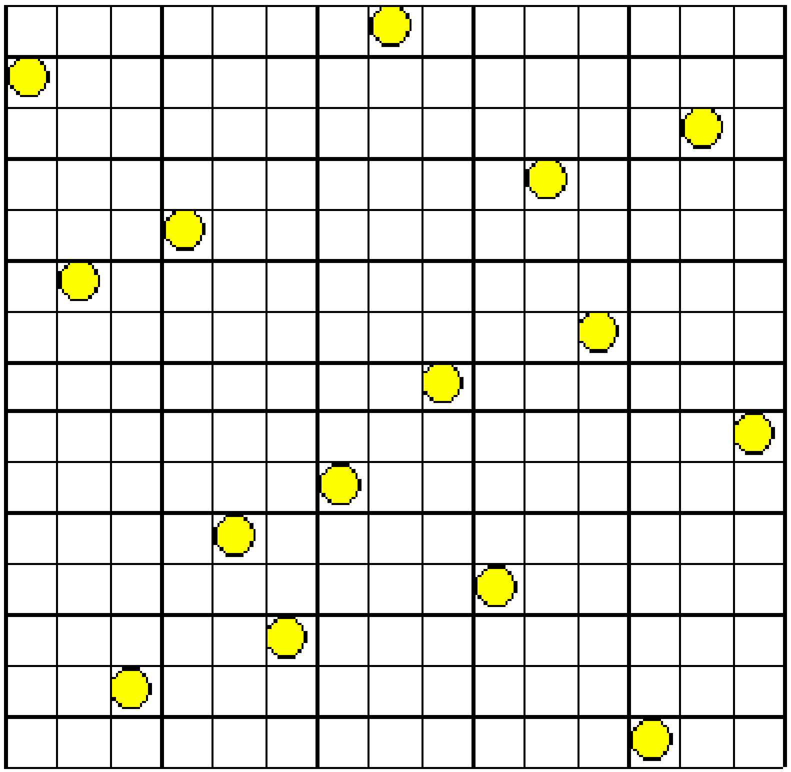


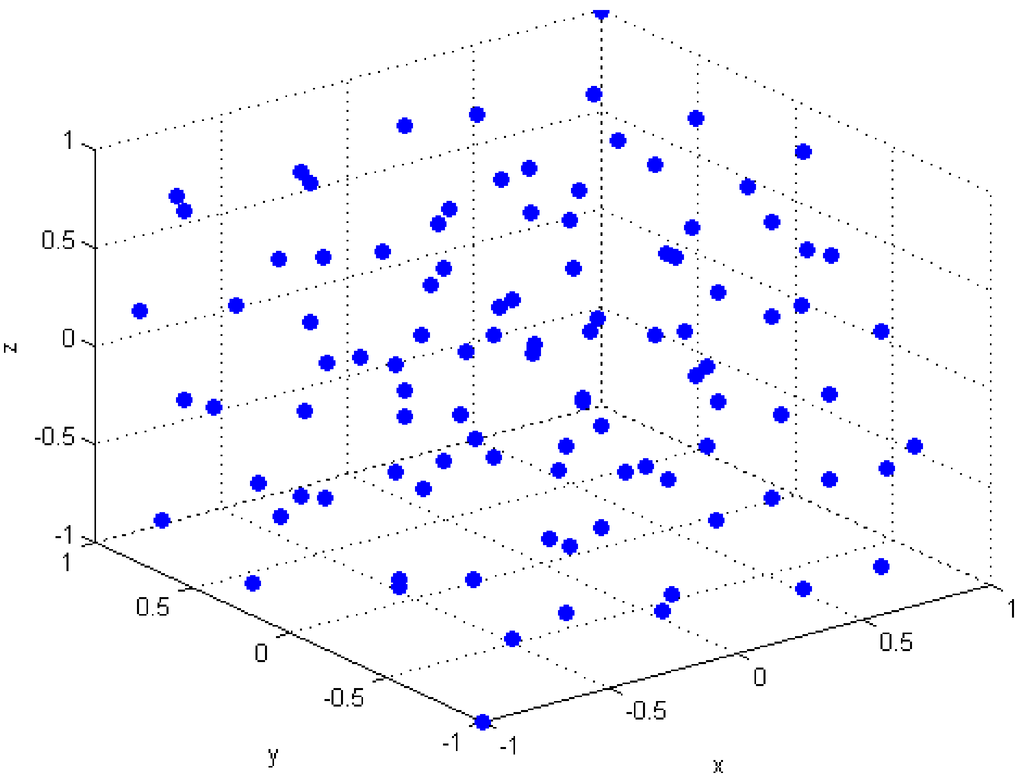





# *Is a small LUT of #500 samples sufficiently covering the parameter space?*

## Latin Hypercube Sampling (LHS)





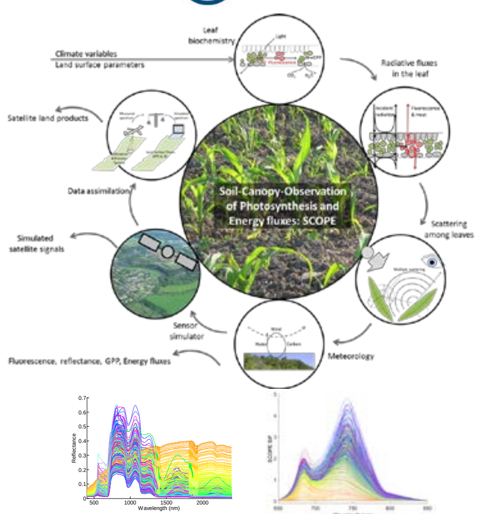
# Experimental setup: emulating SCOPE

**Experimental setup:**

- SCOPE: LUT **500#** @ 1 nm; 400-2500 nm; **8 variables, *R* and *SIF* outputs**
- 6 machine learning methods tested: **RF, SVR, KRR, NN, GPR, VH-GPR**
- **# 10, 20, 30, 40** components tested; **70/30%** Training/Validation (T/V)
- **LUT of # 500, 1000** samples

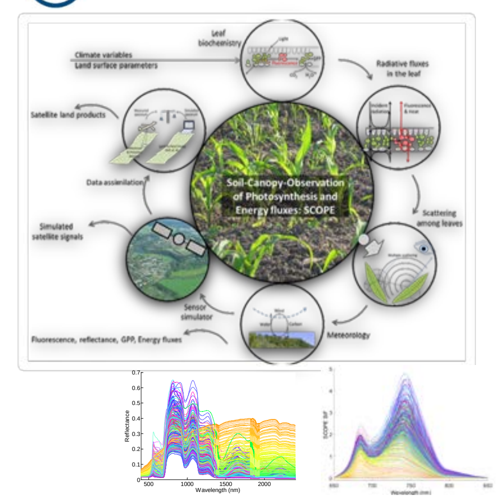
- N (1-3)
- LCC (0-100)
- Cw (0-0.5)
- Cdm (0-0.5)
- Cs (0-0.3
- LAI (0-10)
- Hc (0-2)
- SMC (0-0.7)

**SCOPE: 37 min (500#)**

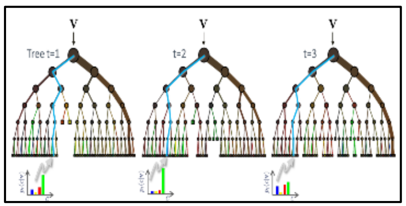


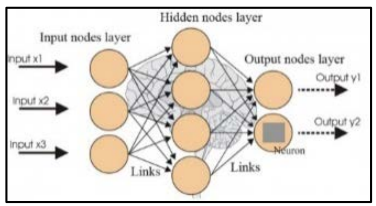

**Emulator**

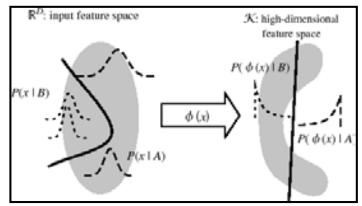
**Emulated SCOPE**

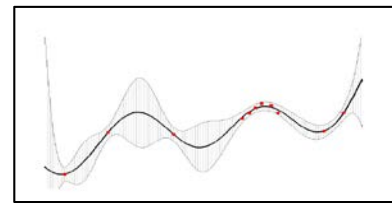


# Role of machine learning



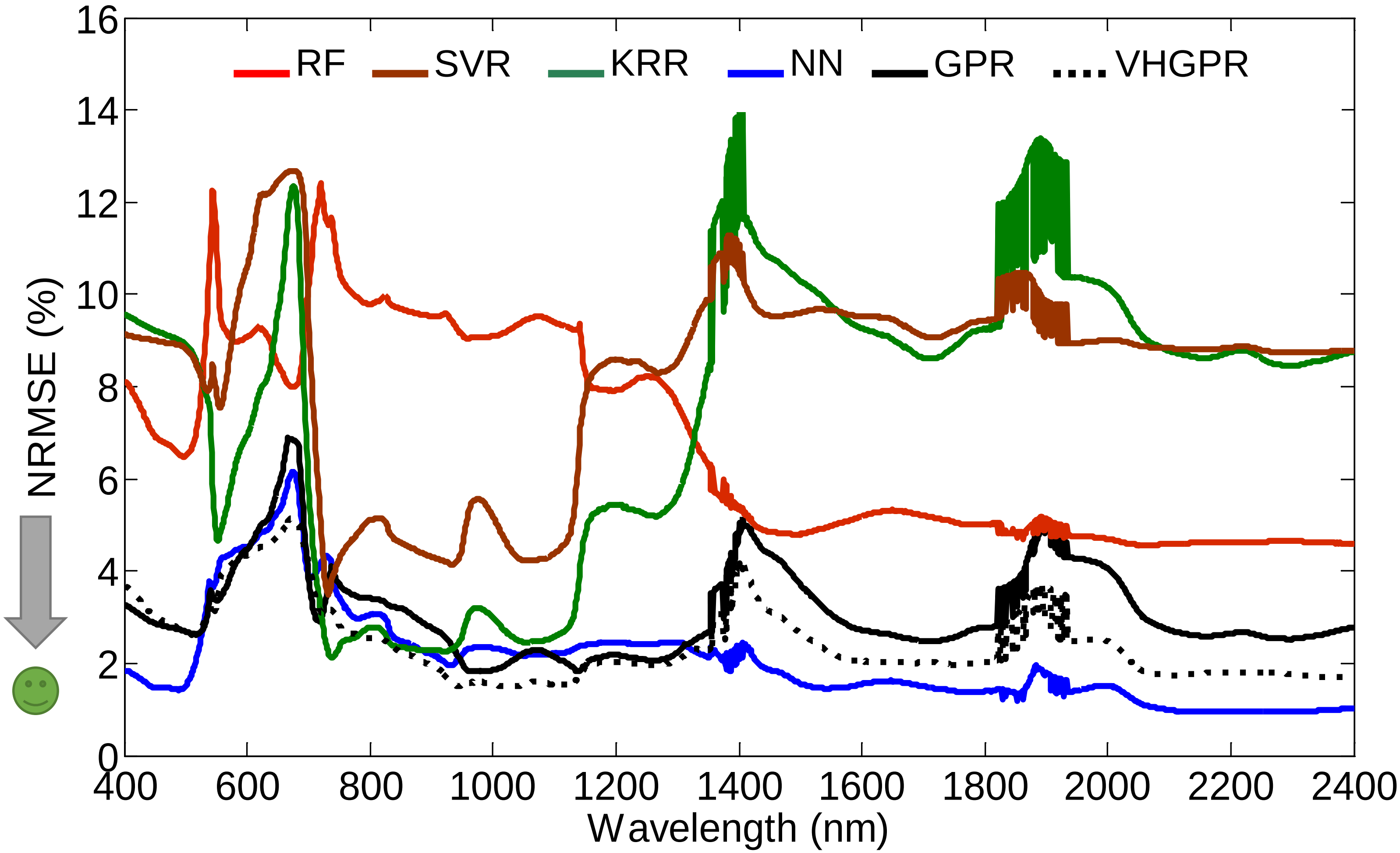



- **Random Forests (RF)**
- **Neural Networks (NN)**
- **Support vector regression (SVR)**
- **Kernel Ridge Regression (KRR)**
- **Gaussian Processes Regression (GPR)**
- **Variational Heteroscedastic GPR (VHGPR)**

# Validation SCOPE *R* emulation

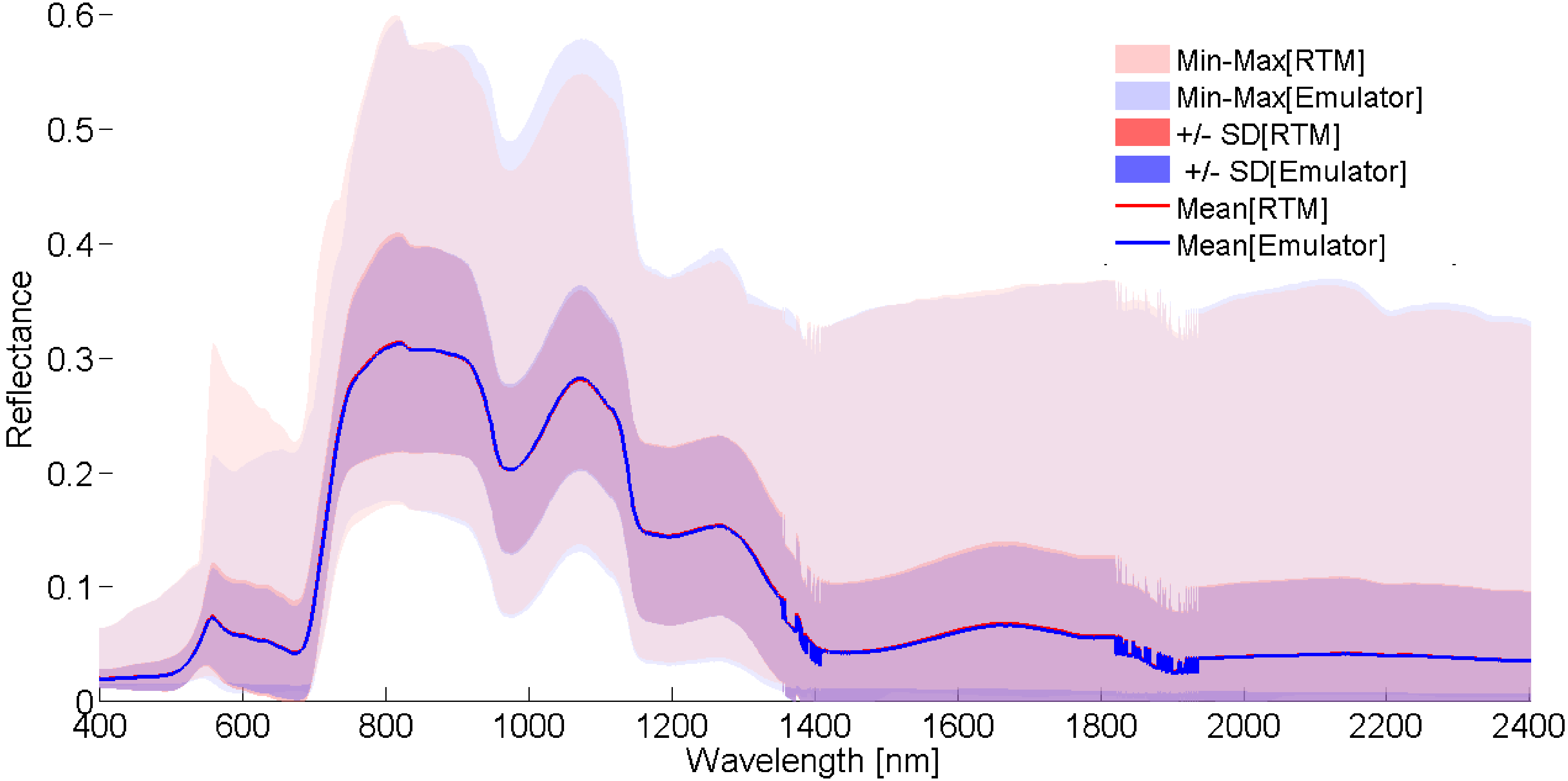
## 1) Role of machine learning

Relative errors for **30%** validation data (#150) with **20 PCA.**

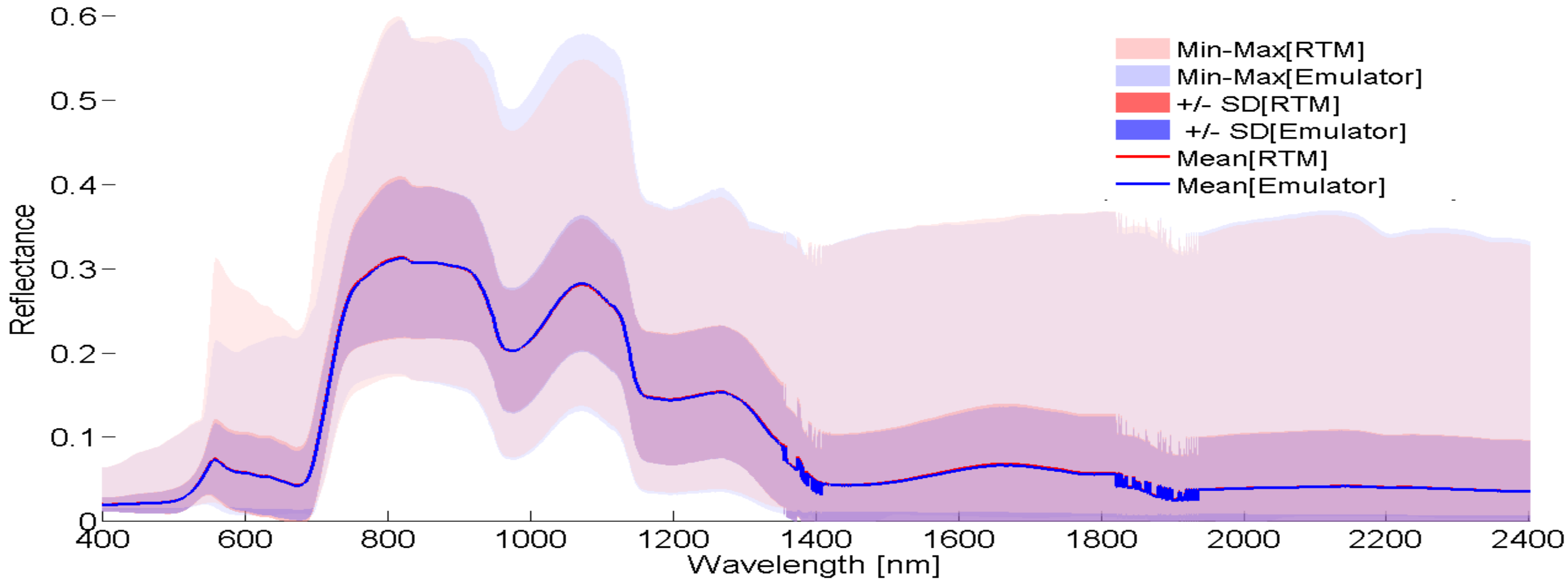



Here, **NN** best performing (<3%). *Let's have a closer look.*

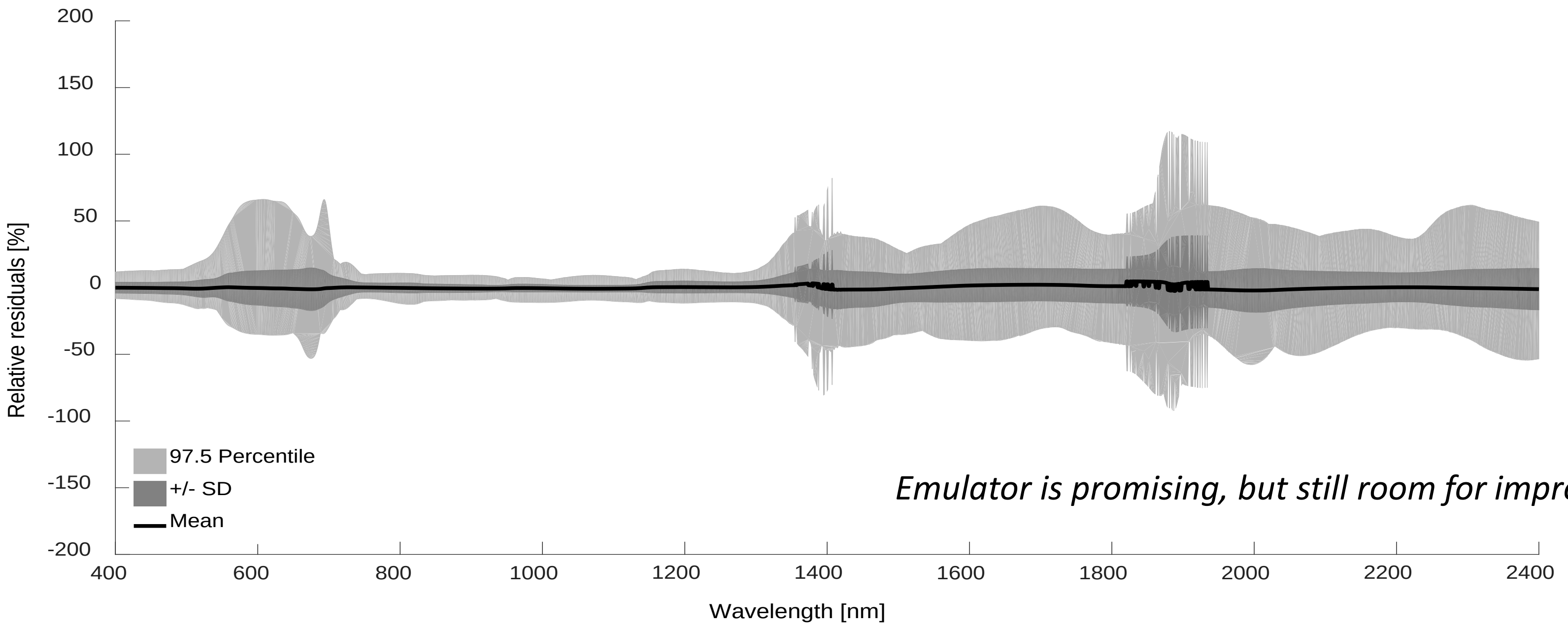
# NN emulator vs RTM validation data: overview stats (30%: 150#)




**The mean and SD closely matching, however min-max boundaries poorer.**

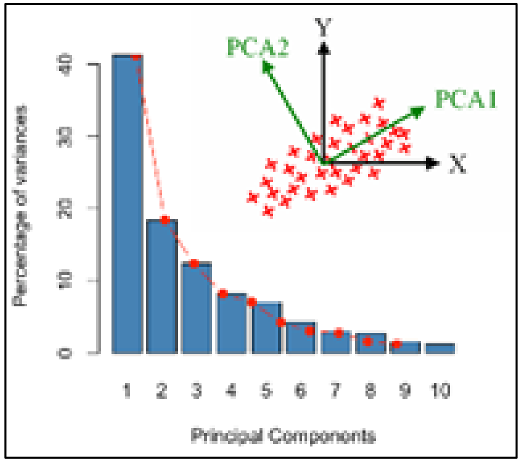


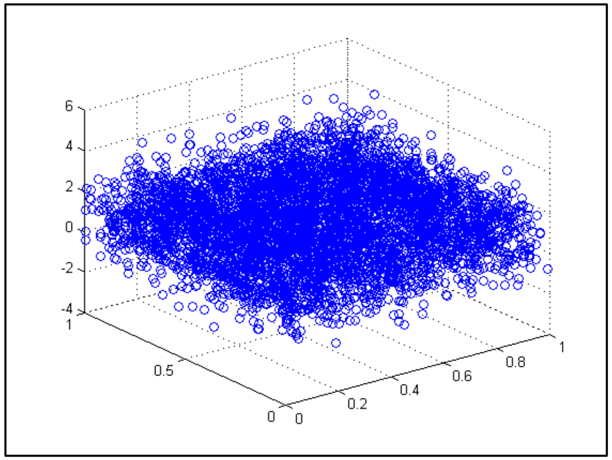

## Statistics of relative residuals:




*Emulator is promising, but still room for improvement?*

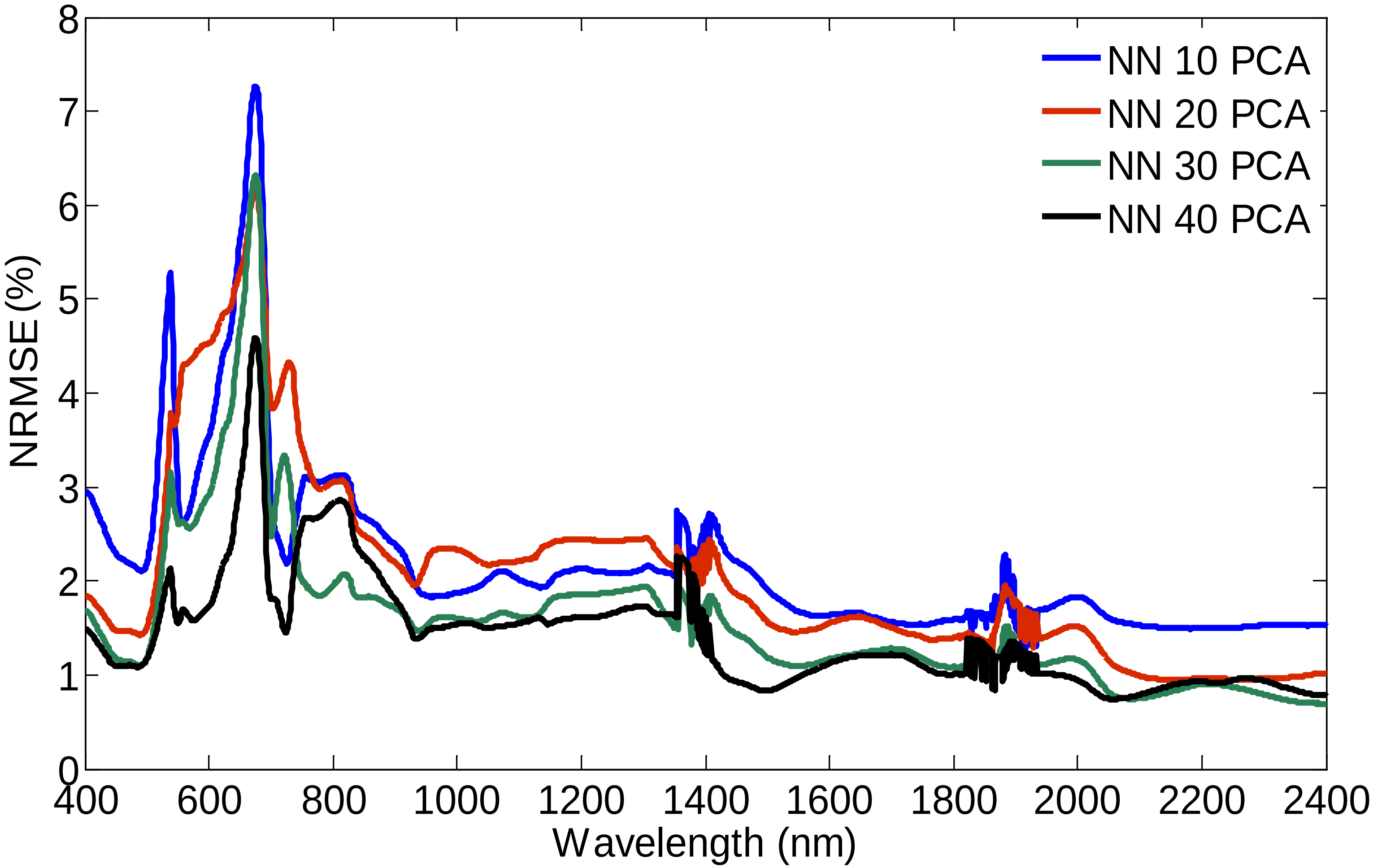
# Role of PCA components & LUT size



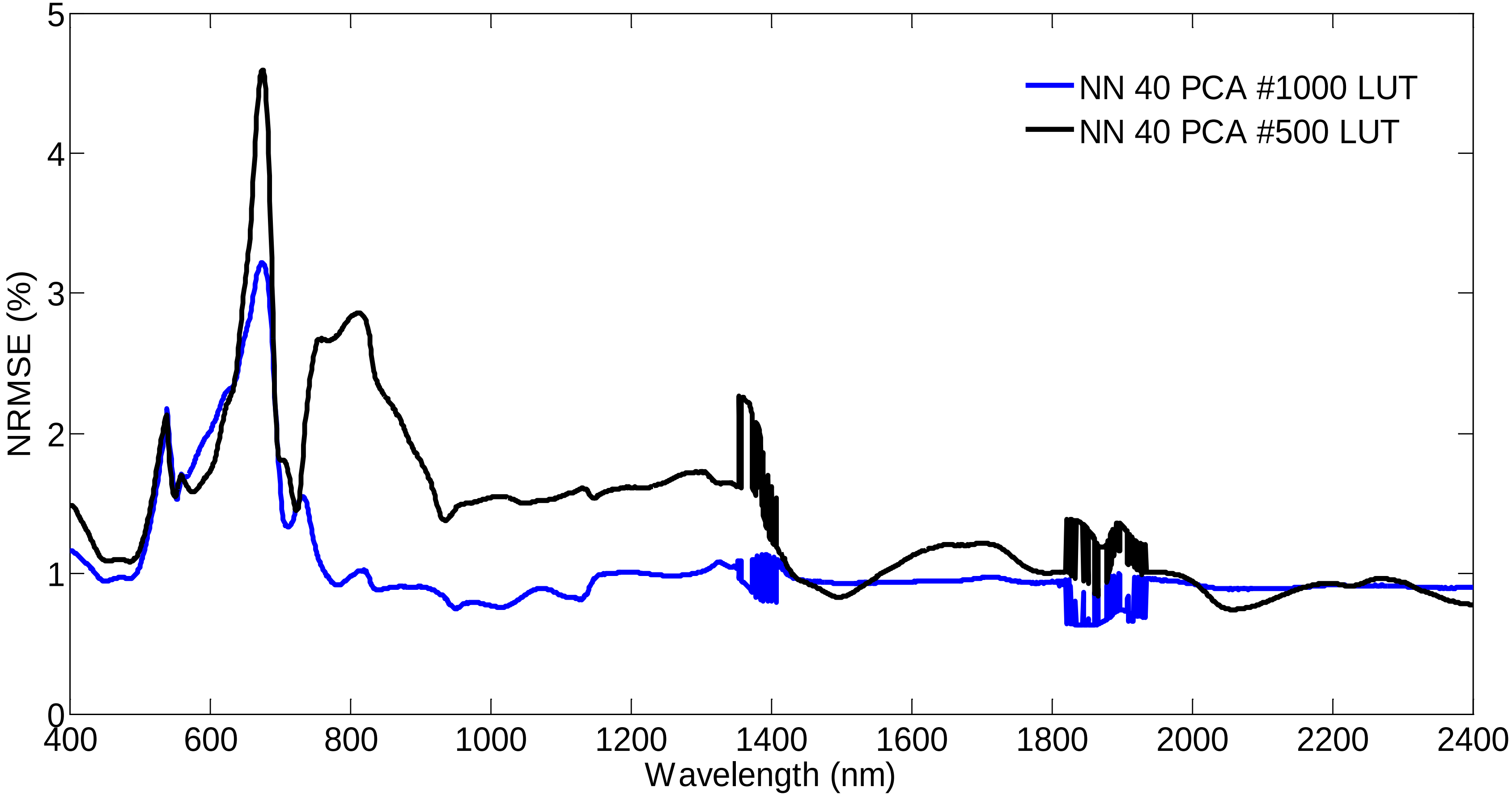
# 2) Role of #PCA components:

10, 20, 30, 40 PCA



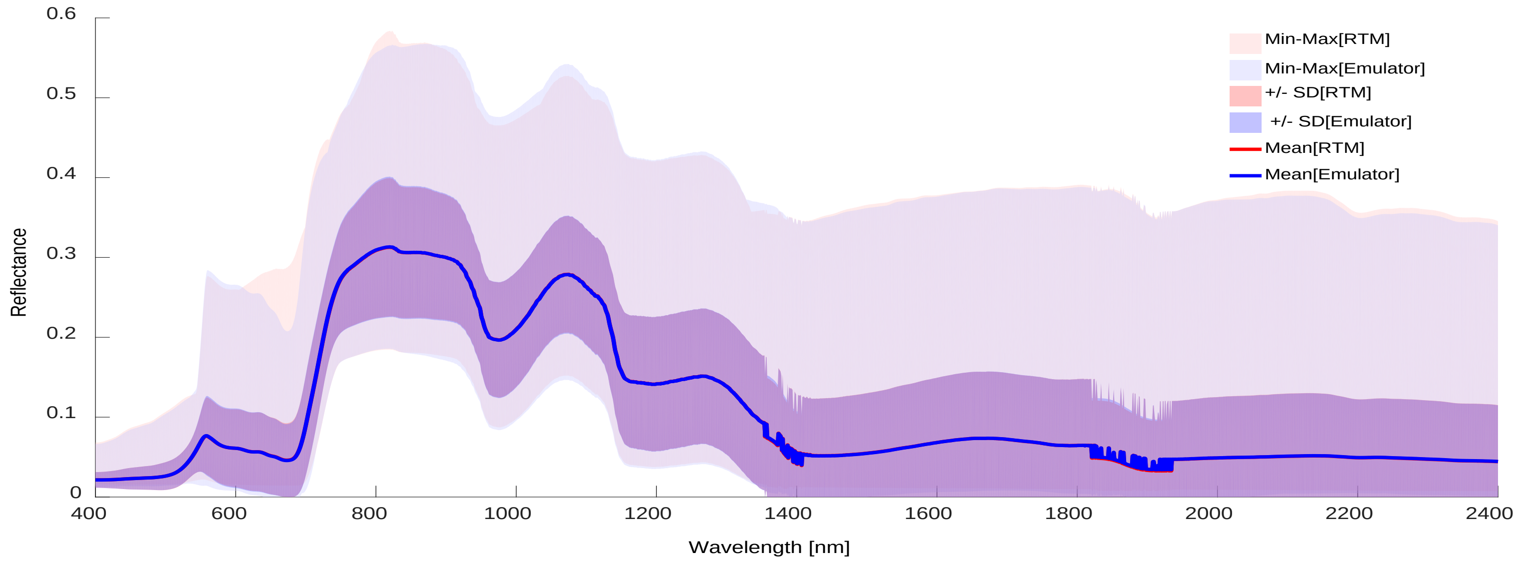

The # of components has a considerable impact on the accuracy: **40 components bring errors down to < 2%** *but slows down processing a bit*.

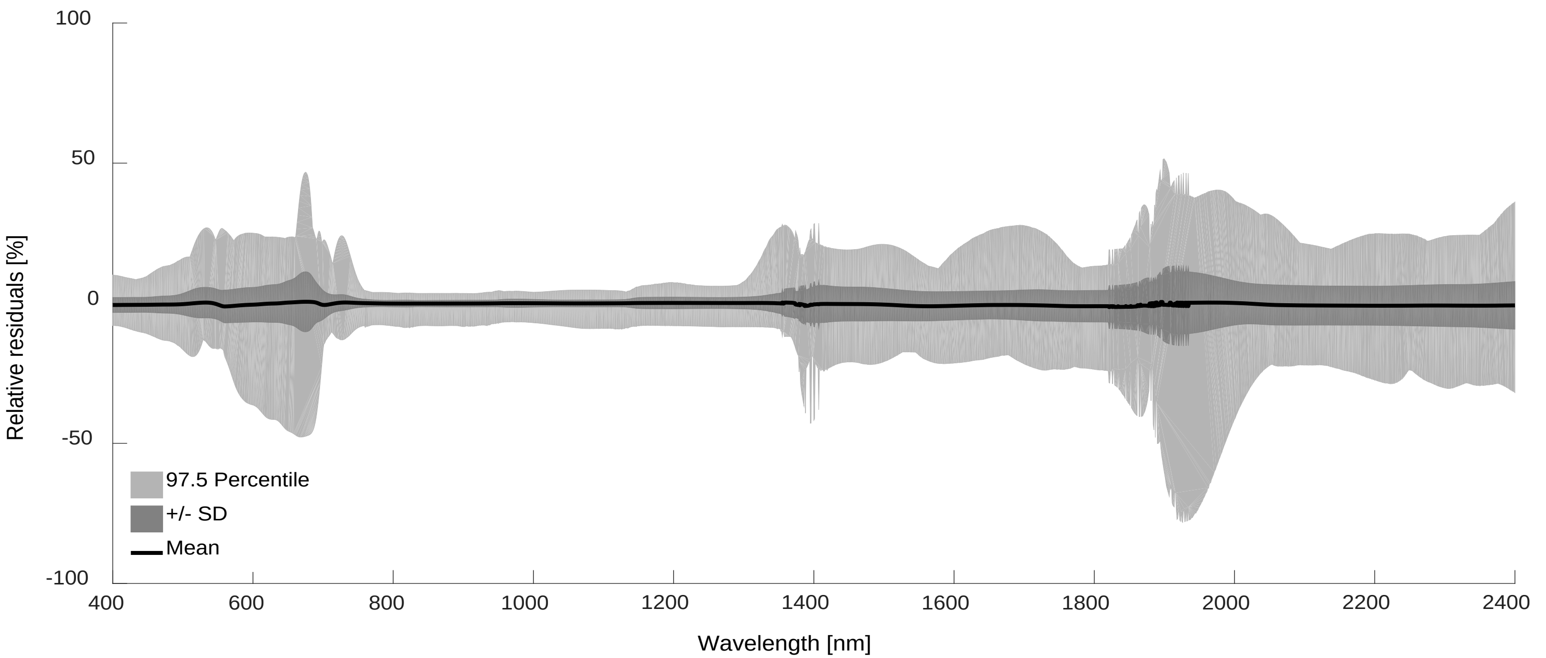
# **3) Role of LUT size (40 PCA):** #500, #1000




**Larger LUT improves accuracy** *but takes longer (69 min) and slows down training: ~42 min.*

# Best performing emulator (NN, #1000, 40PCA)

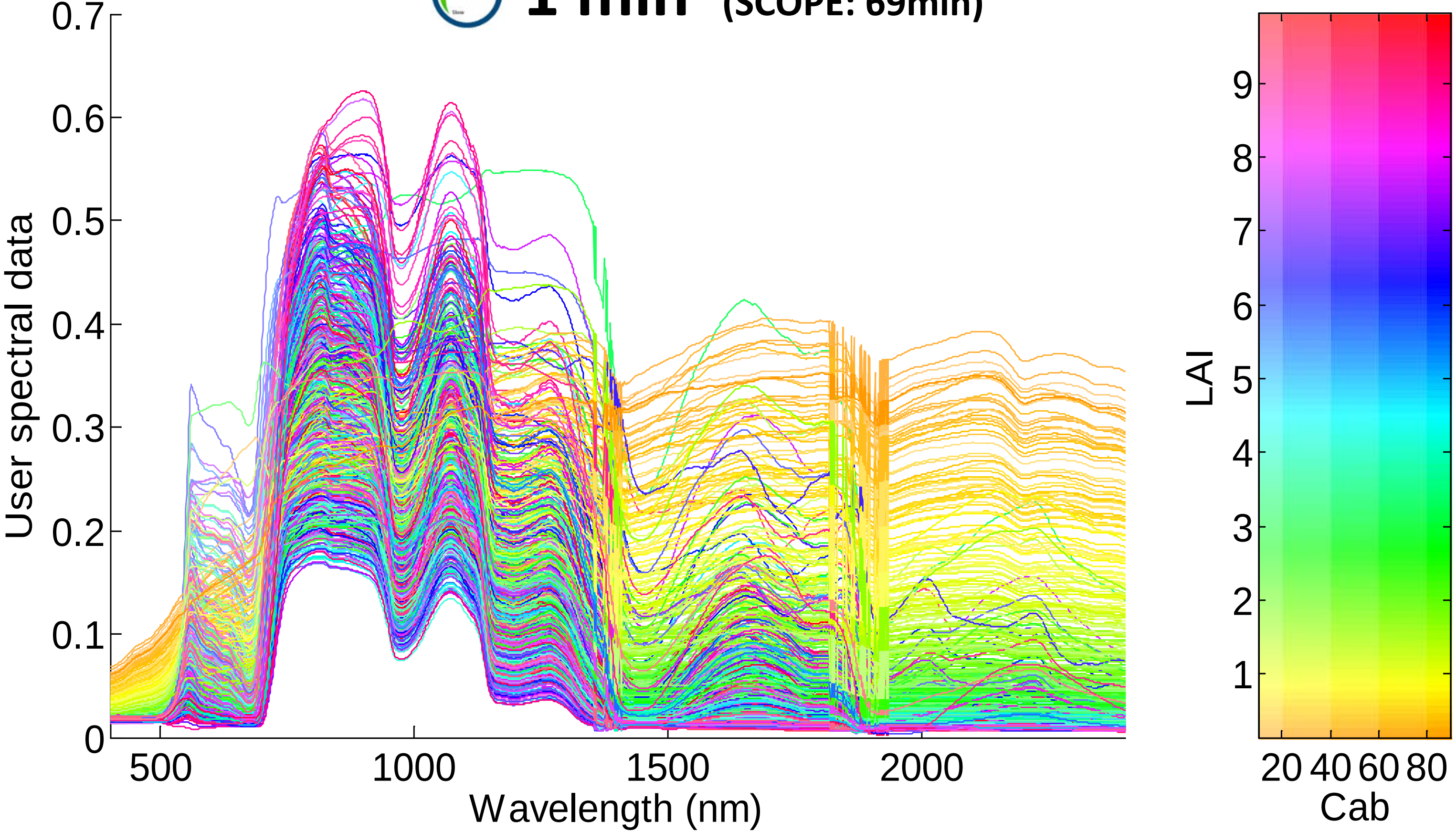


# Generation of **1000# random spectra** by best emulator (NN-40PCA)

**1 min** **(SCOPE: 69min)**

- N (1-3)
- LCC (0-100)
- Cw (0-0.5)
- Cdm (0-0.5)
- Cs (0-0.3
- LAI (0-10)
- Hc (0-2)
- SMC (0.-0.7)

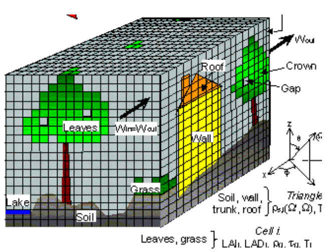



*The emulator covers the full parameter space.*

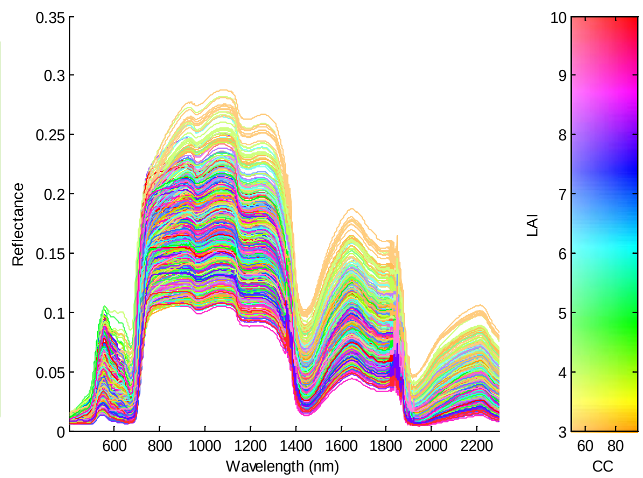
# Emulating an advaced3D RTM: DART

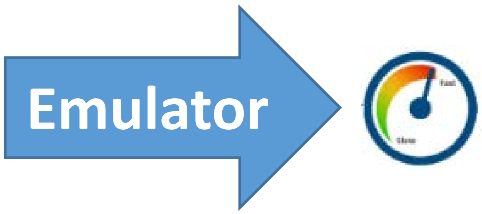
**<u>Experimental setup:</u>**

- DART: LUT**1000#** @ 1 nm; 400-2300 nm; 7 variables; 70/30% T/V
- 3 MLRAs tested: **KRR, NN, GPR**
- Various # PCA components tested (5, 10, 20, 30): 20 best




- N (1.9-2.8)
- LWC (0.01-0.04)
- DMC (0.005-0.04)
- Carc (2.5-20)
- LCC (5-90)
- CC (50-95)
- LAI (3-10)
- TOPO (1-4)

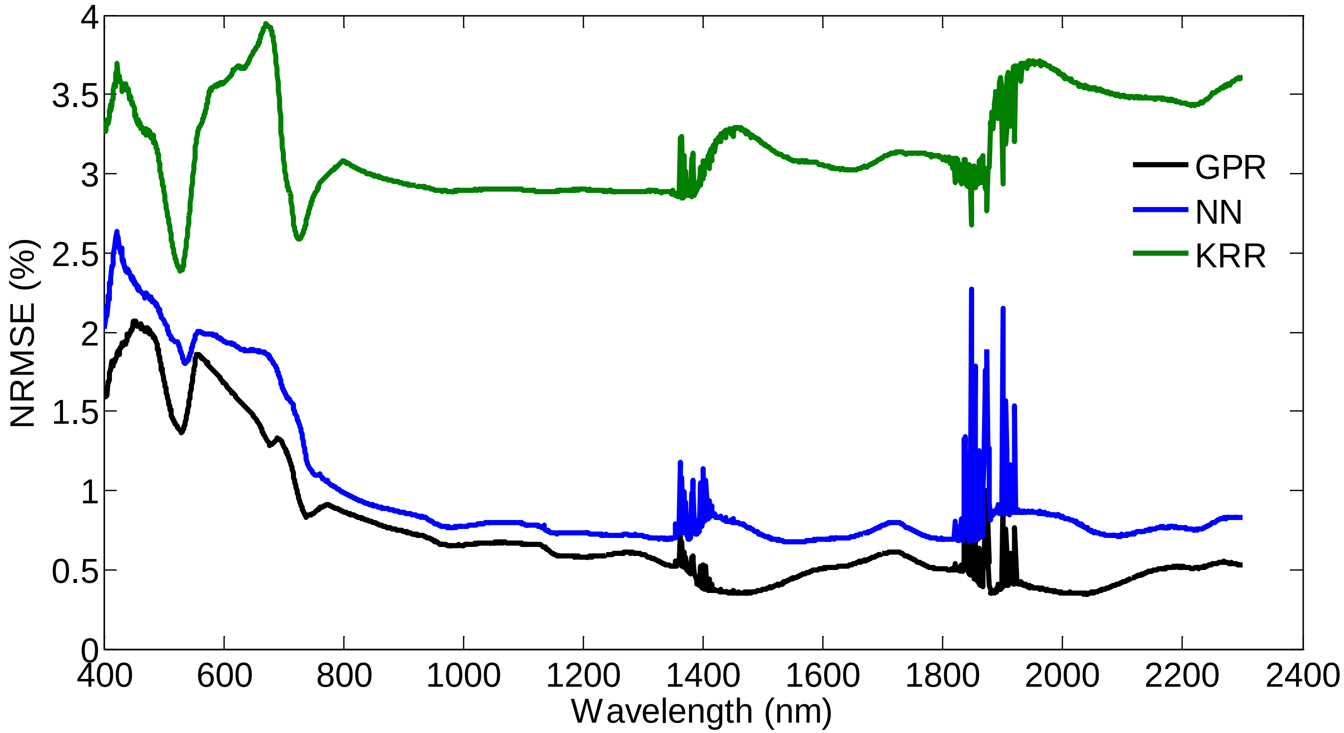


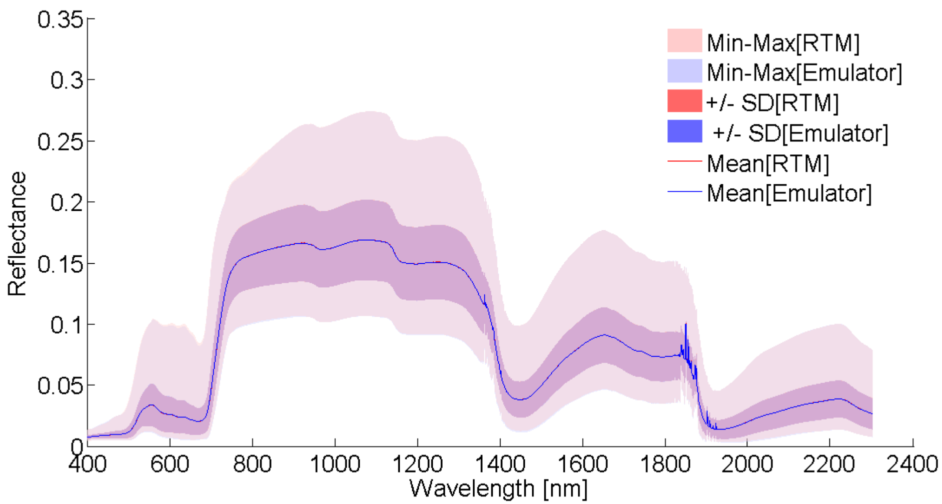

*This is a narrower dataset for training.*

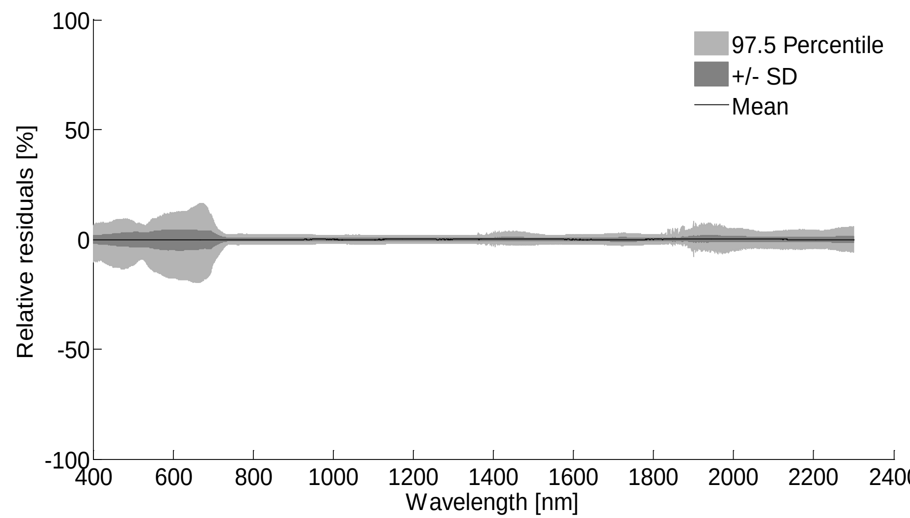
CzechGlobe ☺ *Thanks to Ruzena Janoutova, Lucie Homolova, Petr Lukes*

## 20 PCA

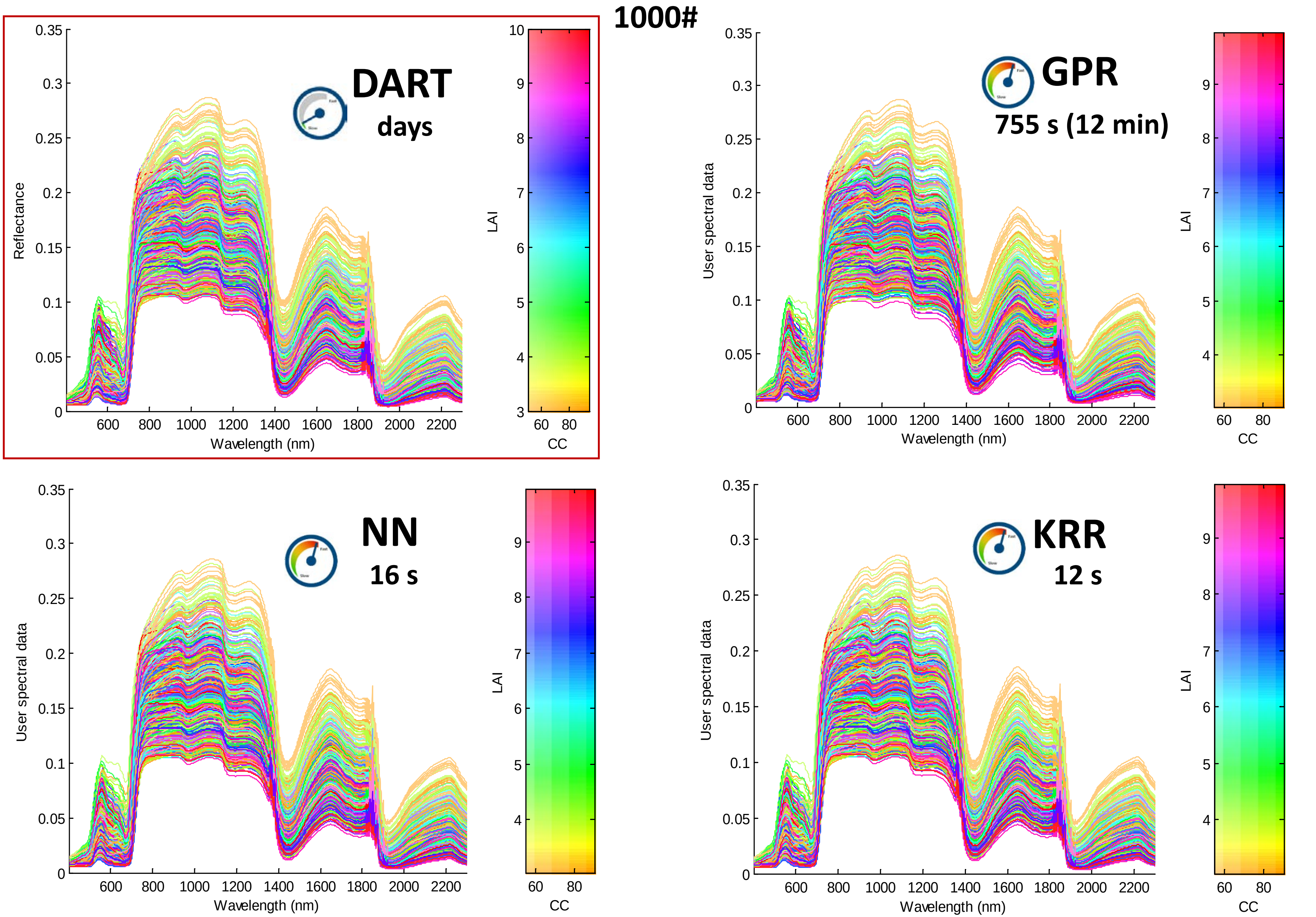



## GPR



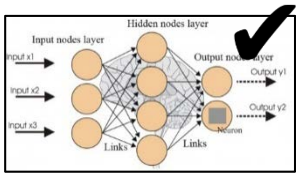

**The narrower DART training set leads to an excellent GPR emulator ☺**

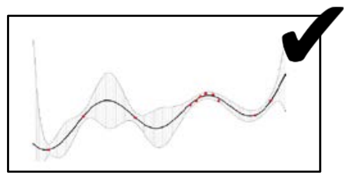



Although GPR provides a slightly better accuracy, because of looping over 20 components, it delivers considerably slower. Thus, **NN preferred** *(despite somewhat poorer accuracies, and 2x longer training time).*

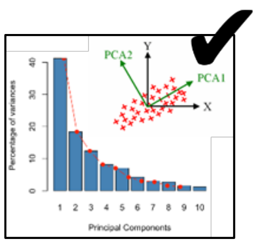
# First conclusions emulation:

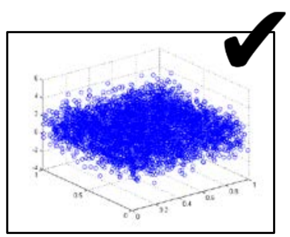
- The type of MLRA most important determining quality of emulation.
  - NN and GPR/VH-GPR best performing.



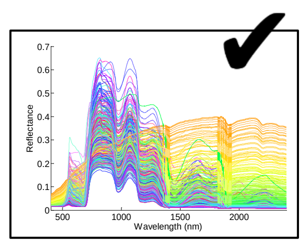



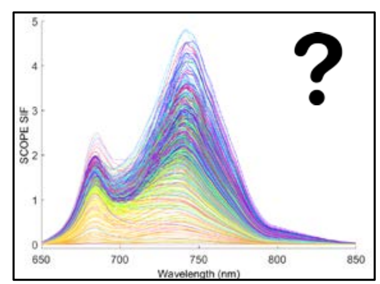
- More components & larger training dataset improve quality.

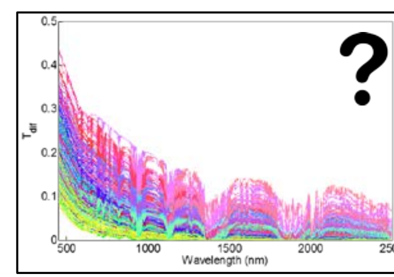


- *Role of data type?*

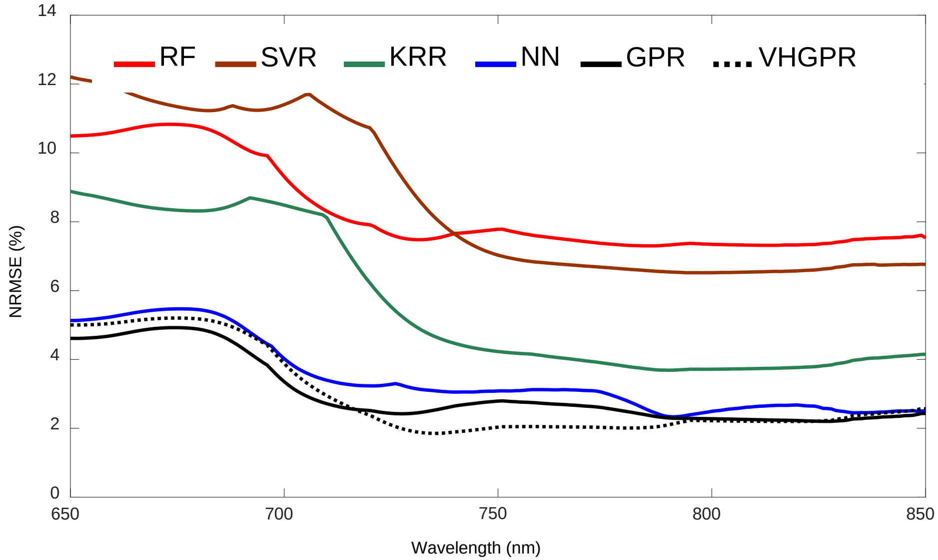


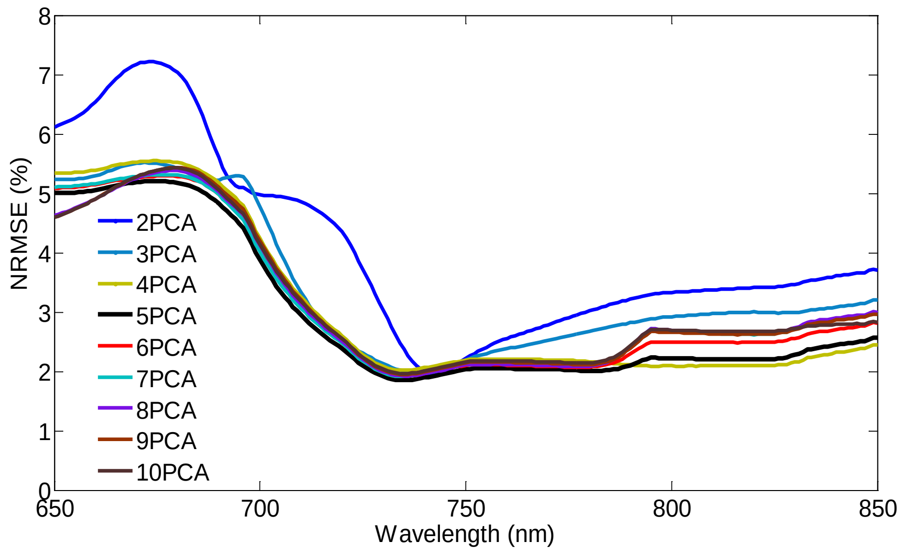



# Validation SCOPE *SIF* emulation

## Role of MLRA (5 PCA):




## Role of components VHGPR:

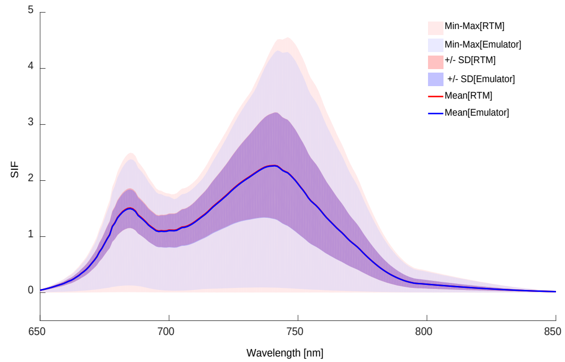


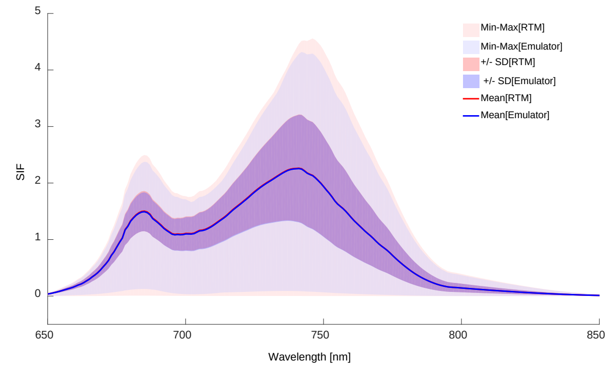

- **GPR/VHGPR** and **NN** again best performing emulators.
- Because a **SIF spectrum is a smooth signal**, 4 components already enough.
- From 8 components onwards, hardly improvements.

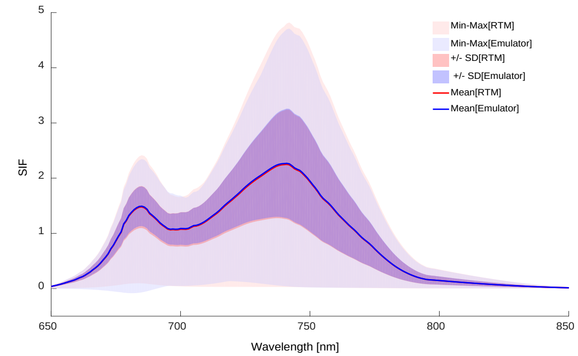
# VHGPR-10PCA

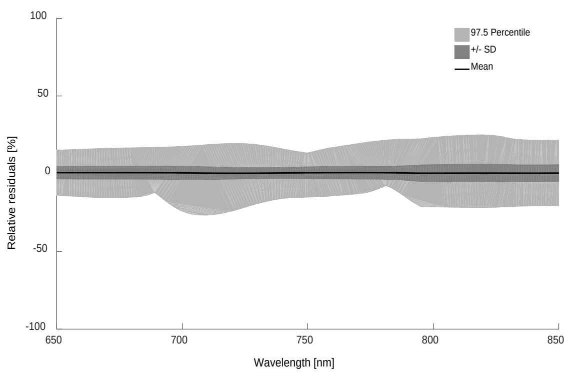
# VHGPR-5PCA

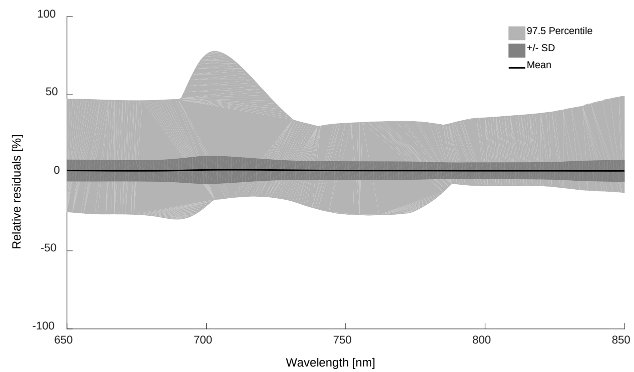
# GPR-5PCA

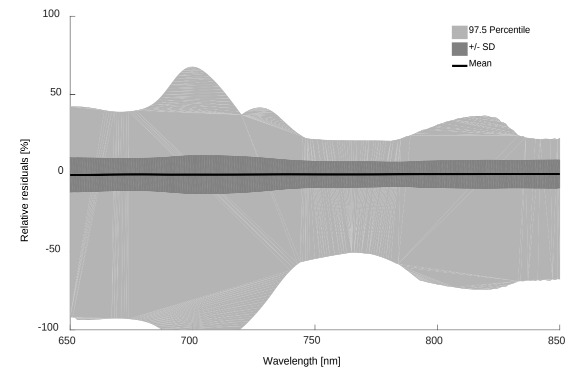


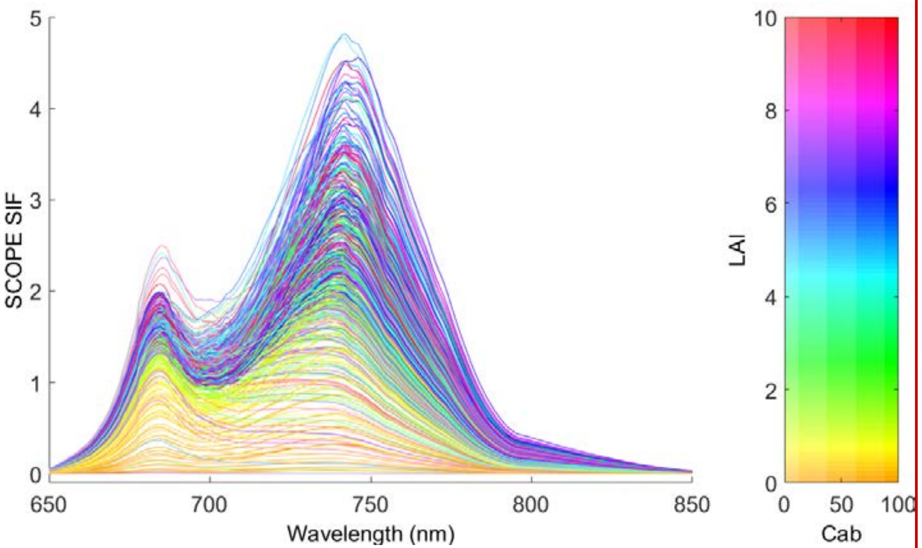



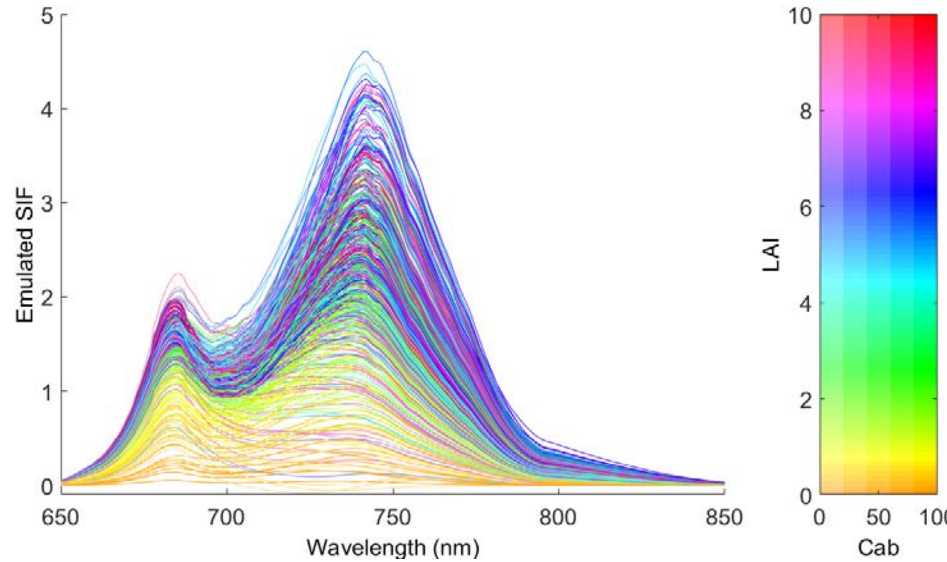




- GPR/VHGPR are not multi-output, but develops models for each component.
- More components somewhat more accurate, but more processing time.

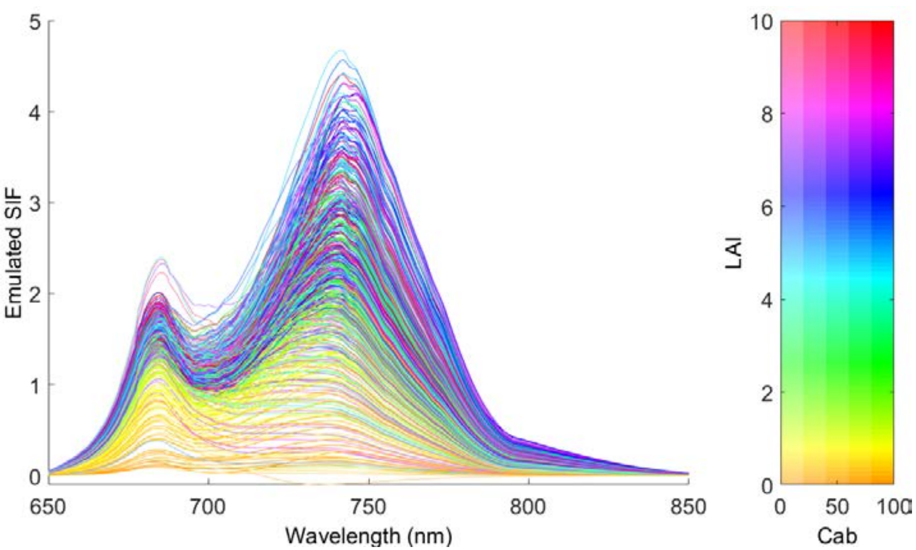
## SCOPE 500# (37 min)



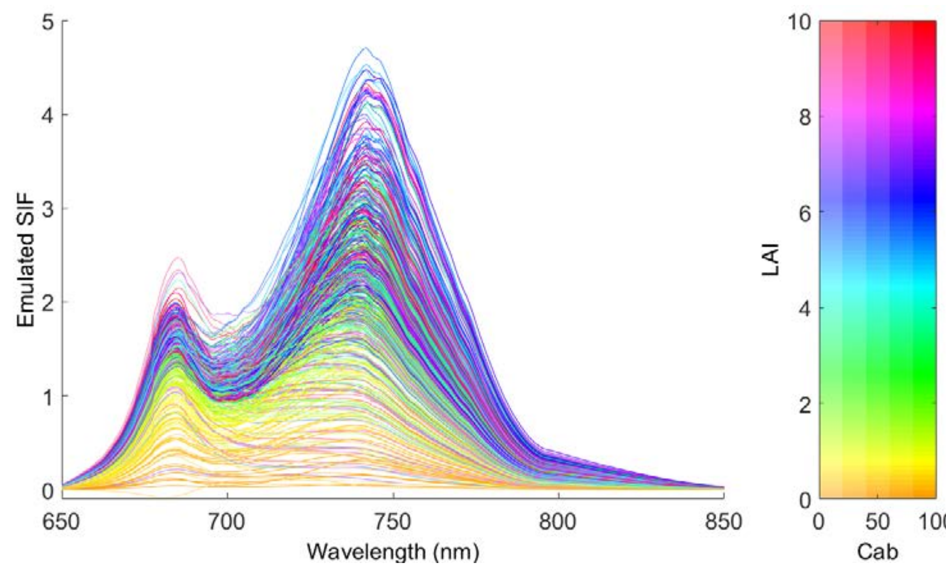

## VHGPR-10PCA 500# (99 s.)




## VHGPR-5PCA 500# (36 s.)




## GPR-5PCA 500# (24 s.)




*NN emulator faster (14 s.) but some negative profiles. Therefore not considered.*

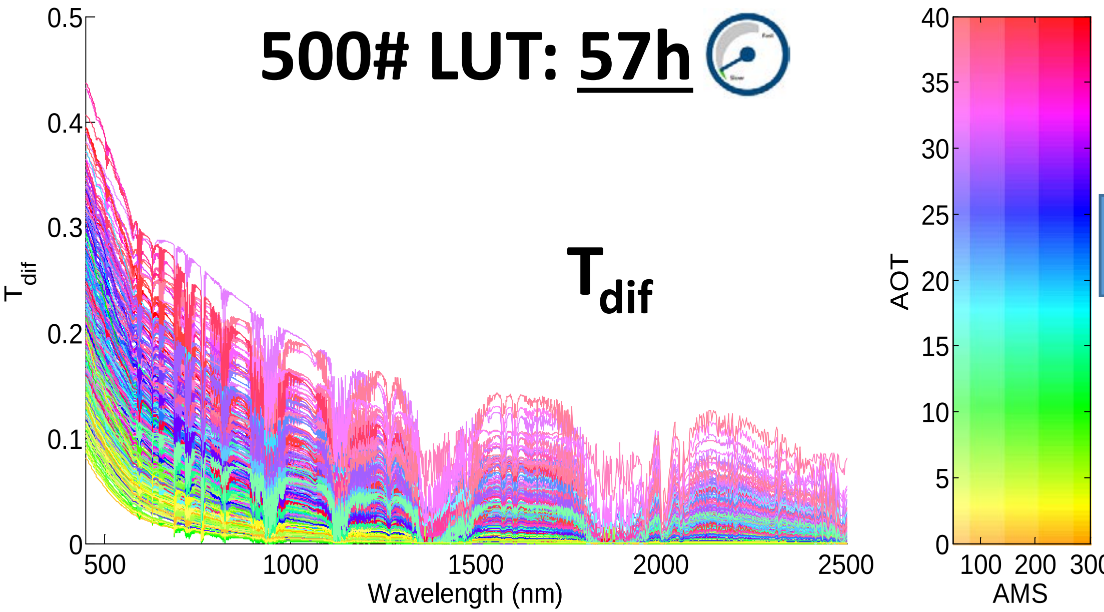
# Emulating an advanced atmospheric RTM: MODTRAN5

## Experimental setup:

- MODTRAN5: LUT **500#**, **3645b**; 400-2500 nm; 70/30% T/V
- 8 variables
- 6 Atmospheric transfer functions: $\mathbf{T_{dif}}$, $\mathbf{T_{dir}}$, $\mathbf{E_{dir}}$, $\mathbf{E_{dif}}$, **S**, $\mathbf{L_0}$
- 3 MLRAs tested: **KRR, NN, GPR**
- **30 PCA**

- VZA (0-55)
- SZA (0-60)
- RAA (0-180)
- ELEV (0-2)
- AOT (0-0.4)
- AMS (0.5-3)
- G (-1 - 1)
- CWV (0-2)

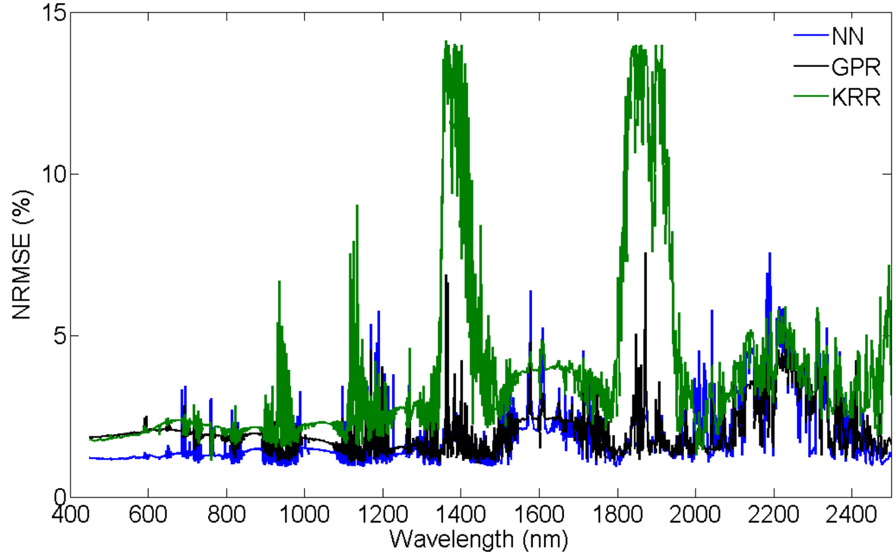
**500# LUT: 57h**

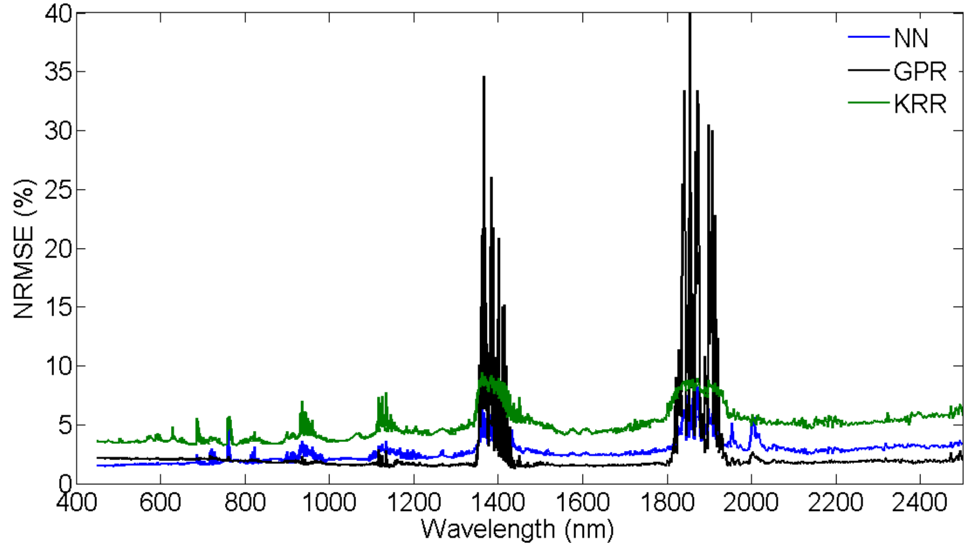



**Emulator**

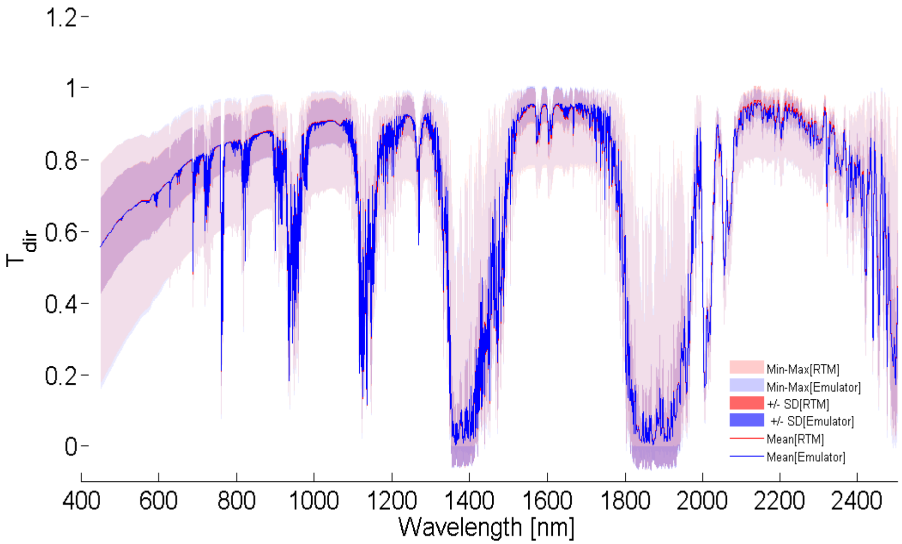
***No longer smooth spectra, but spiky profiles***

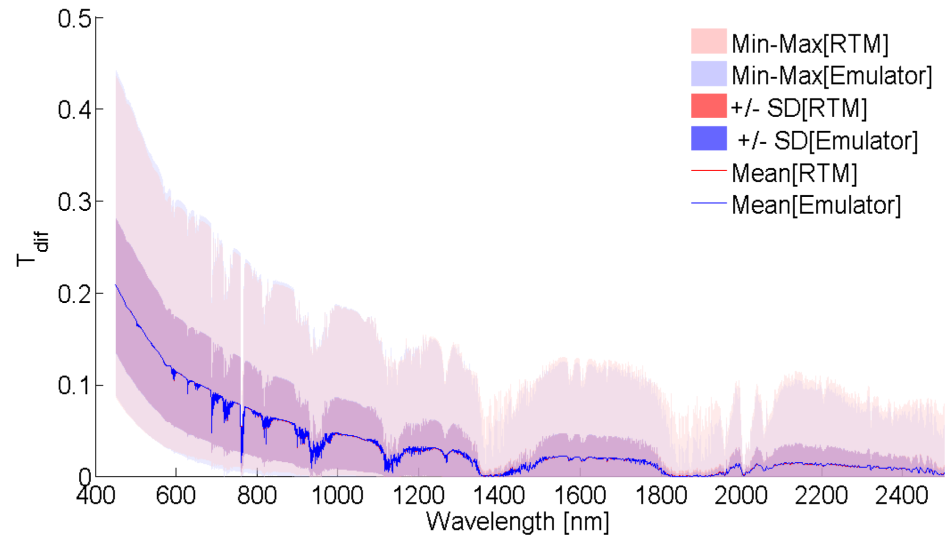
# $T_{dir}$ / $T_{dif}$




**NN Emulator (30 PCA) best performing: <2% errors, with outliers in absorption regions.**

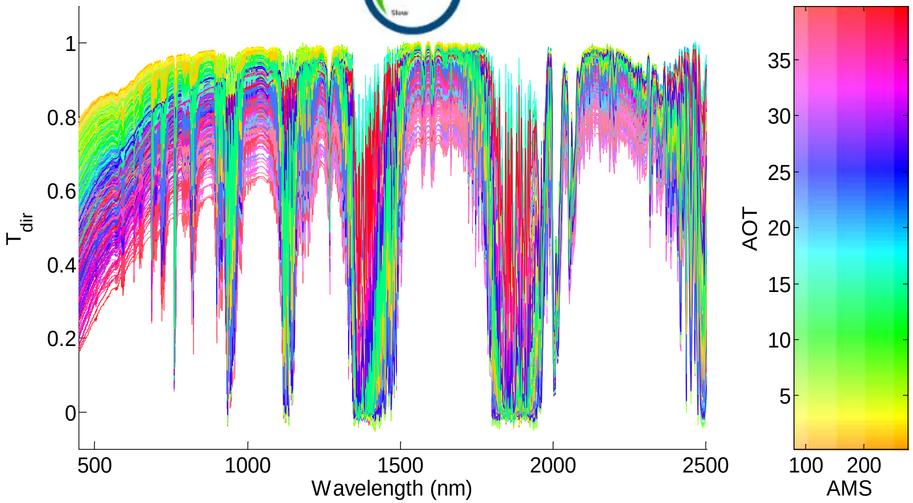


# Emulators: 1000# LUT

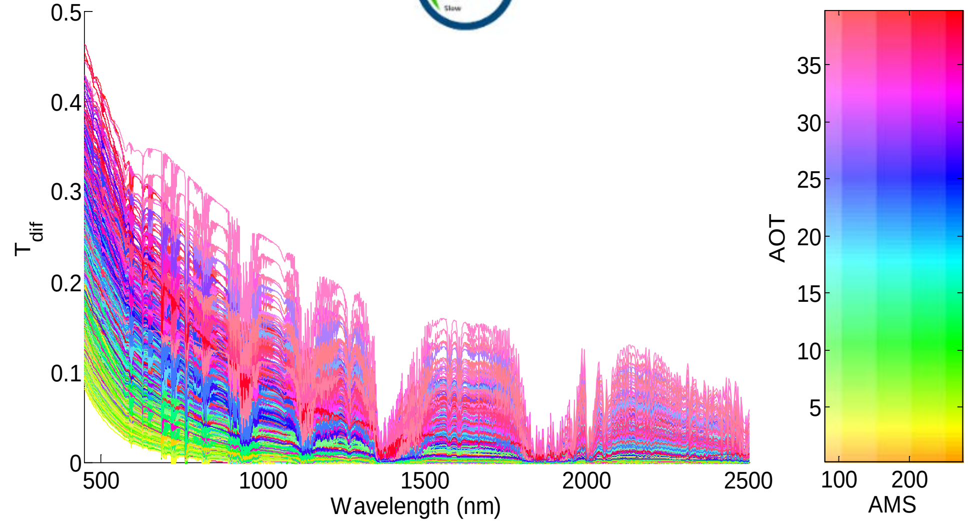
## $T_{dir}$

**NN** **66 s.**



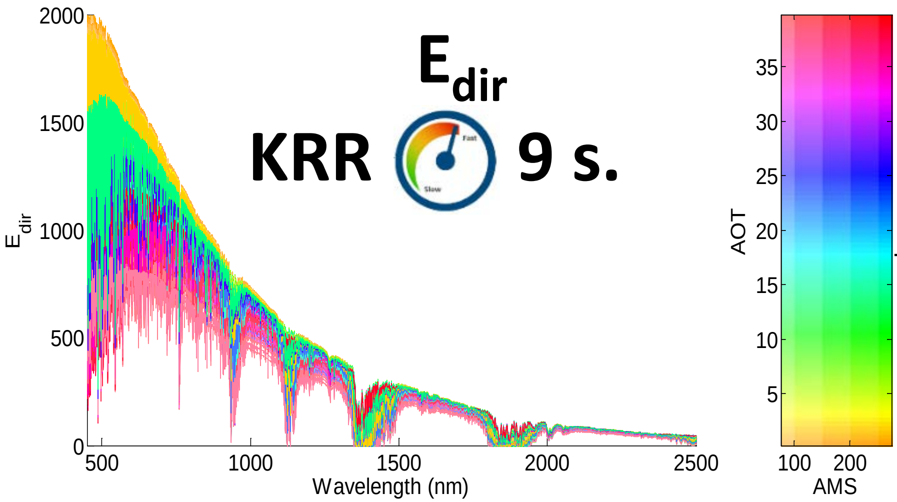

## $T_{dif}$

**NN** **71 s.**



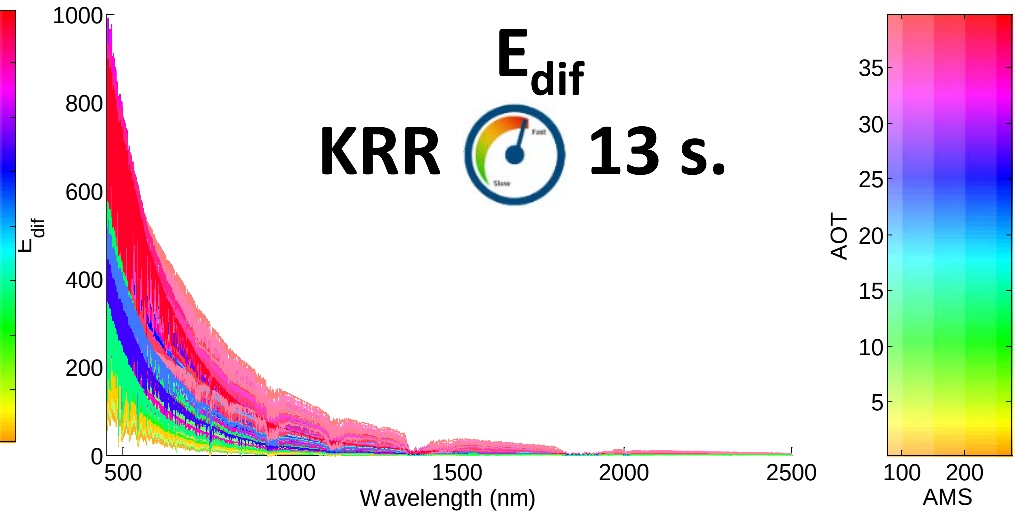

## $E_{dir}$

**KRR** **9 s.**




## $E_{dif}$

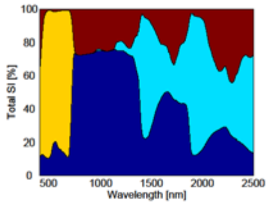
**KRR** **13 s.**




***Using MODTRAN, the same 1000# would take about 5 days.***

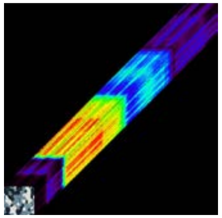
# Applications emulation

✓ Global sensitivity analysis (GSA)

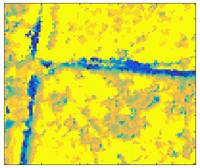



✓ Generation of field data

✓ Scene generation



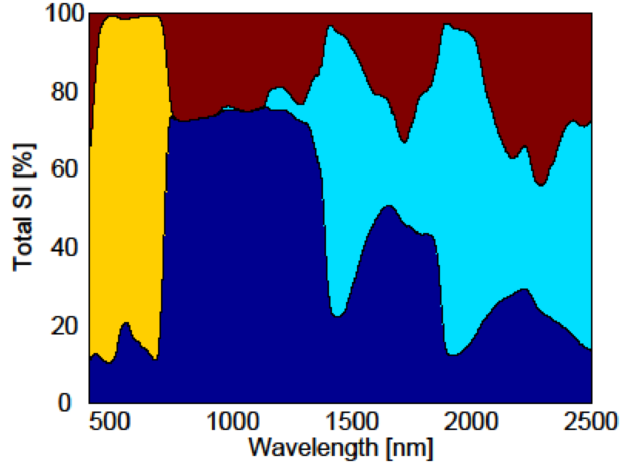
✓ Numerical inversion



# Global Sensitivity analysis

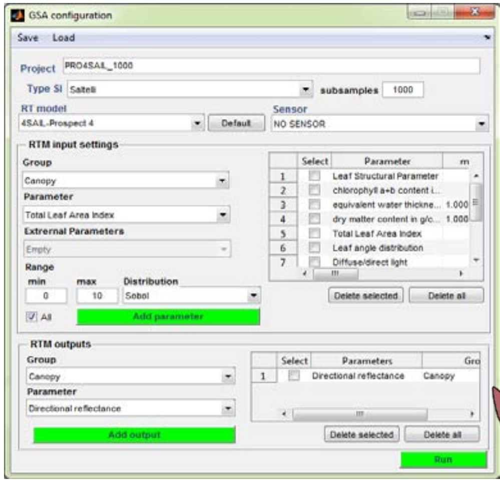
*EARSeL IS 2015, Luxembourg*

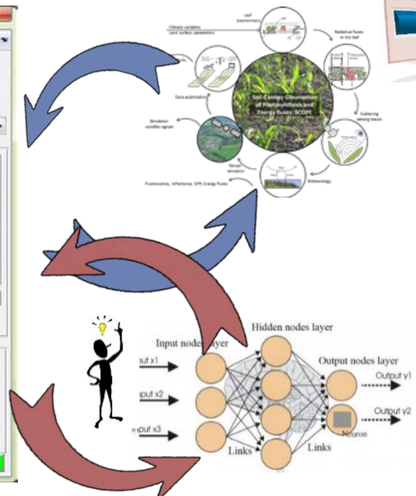
**PROSPECT 4** ~ 2min



# Emulators applied into GSA: PROSPECT-4

**ARTMO's GSA toolbox**

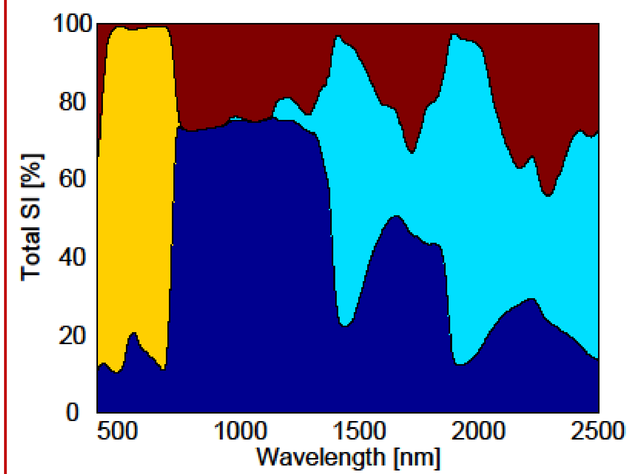


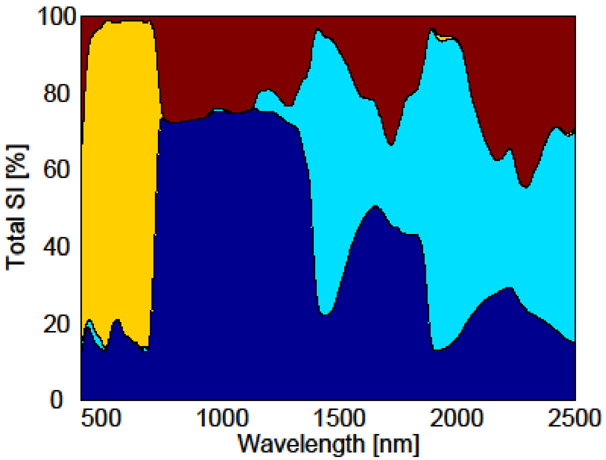


**X 2**

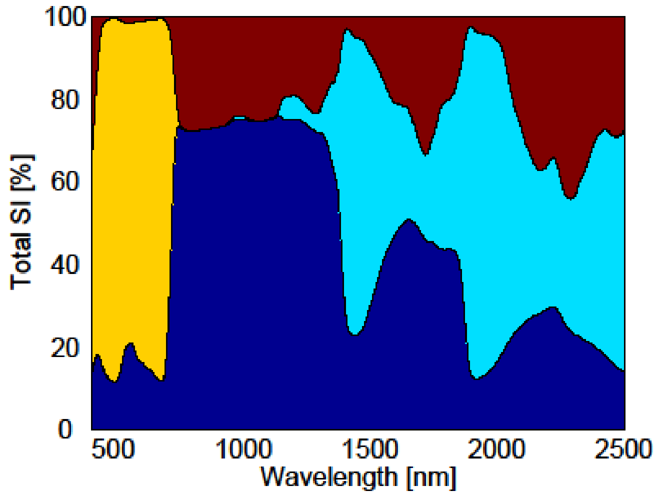

**PROSPECT-4 (122 s)**



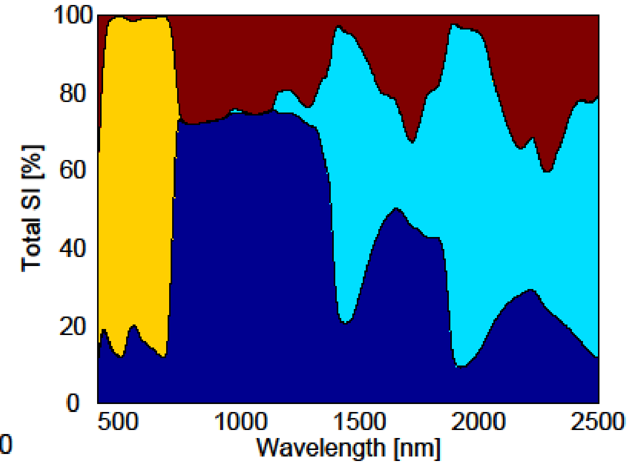

**KRR (55 s)**




**NN (55 s)**



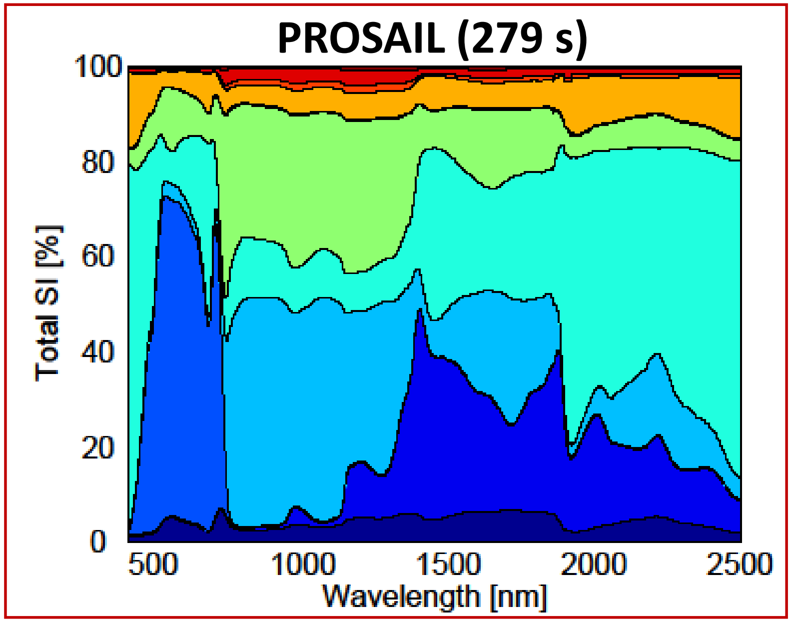

**GPR (58 s)**



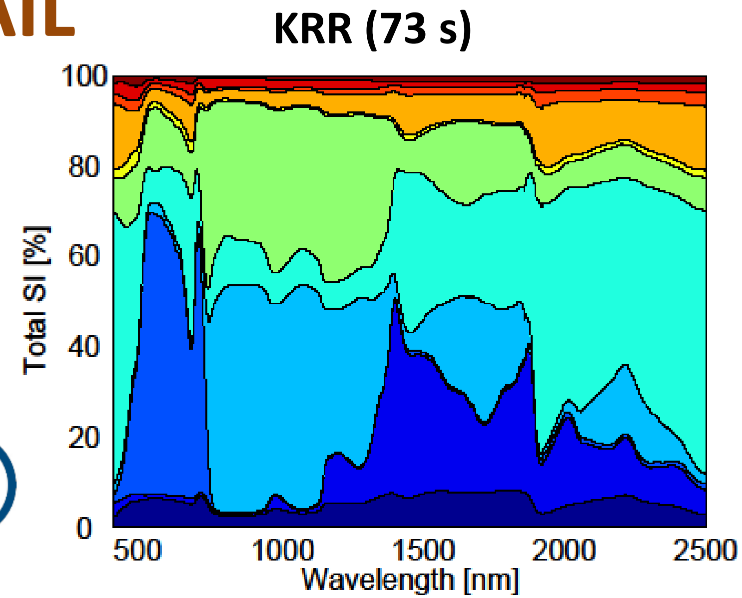


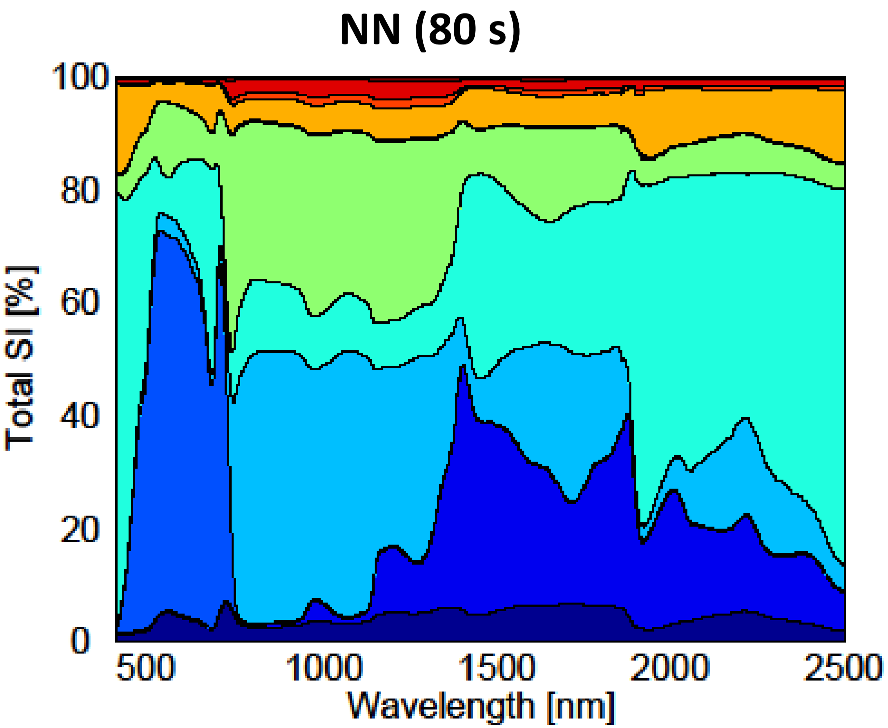
**1000#/variable**

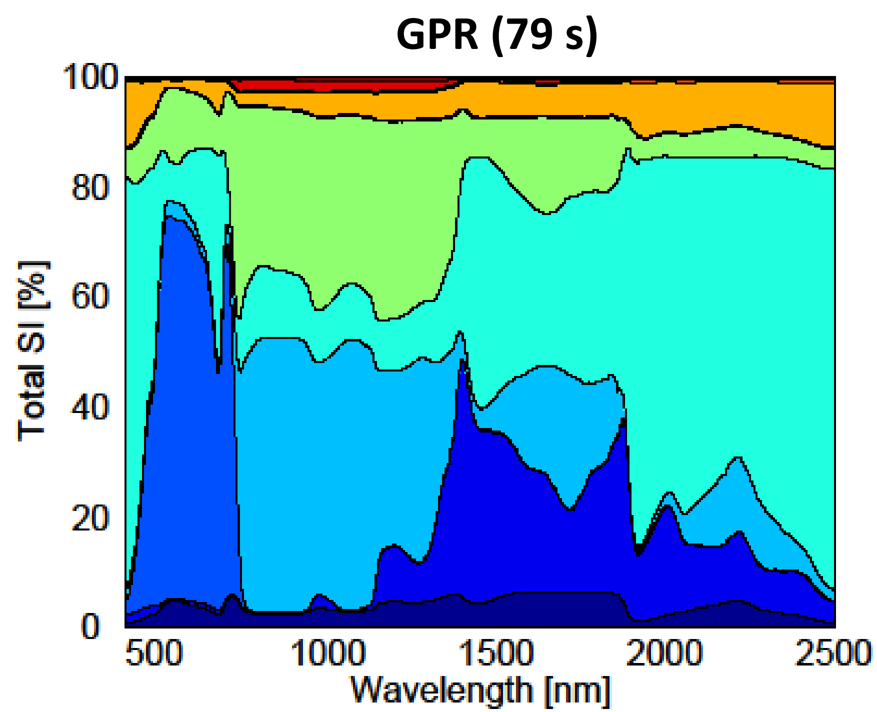
# PROSAIL




X 3.5

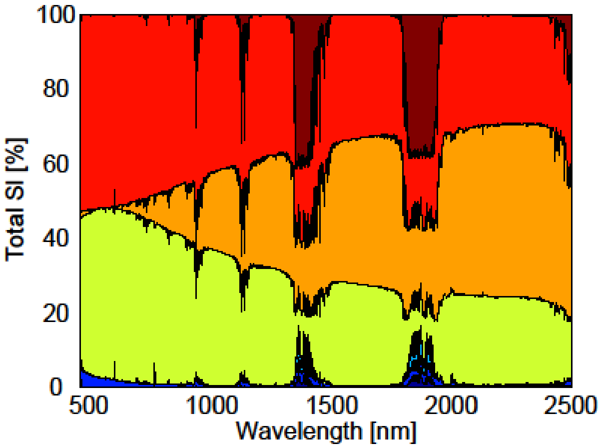

**1000#/variable**

N Cw Cab Cm LAI LAD skyl soil coeff SZA VZA RAA

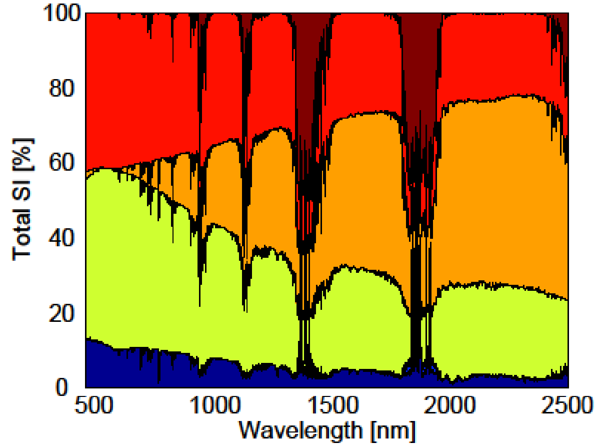
# MODTRAN
# atmospheric transfer functions: $L_{TOA} = L_0 + \frac{(E_{dir}\mu_s + E_{dif})(T_{dif} + T_{dir})\rho}{\pi(1 - S\rho)}$

**X 130,000**

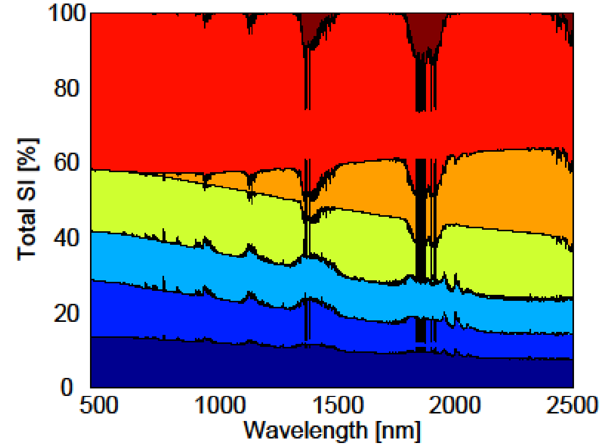
### *$E_{dif}$* (GPR: 121 s)



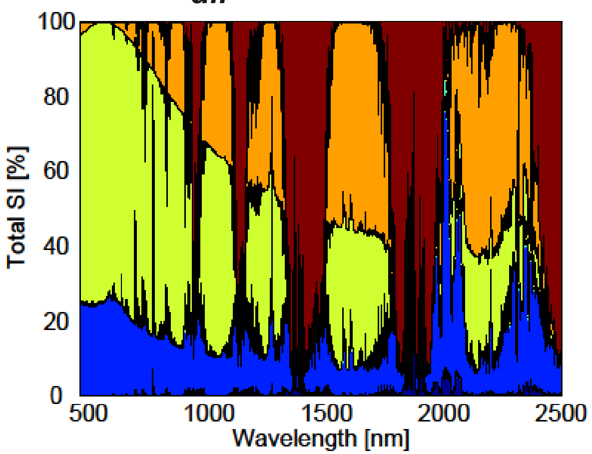

### *$T_{dif}$* (NN: 157 s)



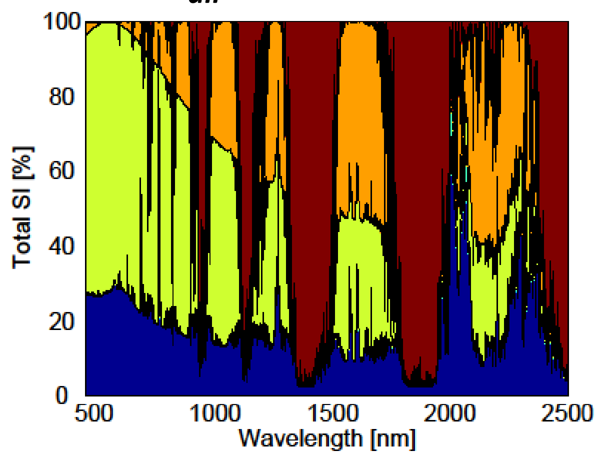

### *$L_0$* (GPR: 121 s)



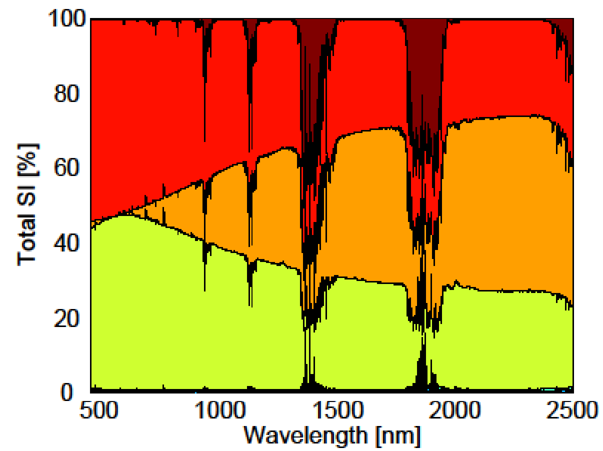

### *$E_{dir}$* (KRR: 101 s)




### *$T_{dir}$* (NN: 151 s)




### *S* (GPR: 166 s)




**1000#/variable**

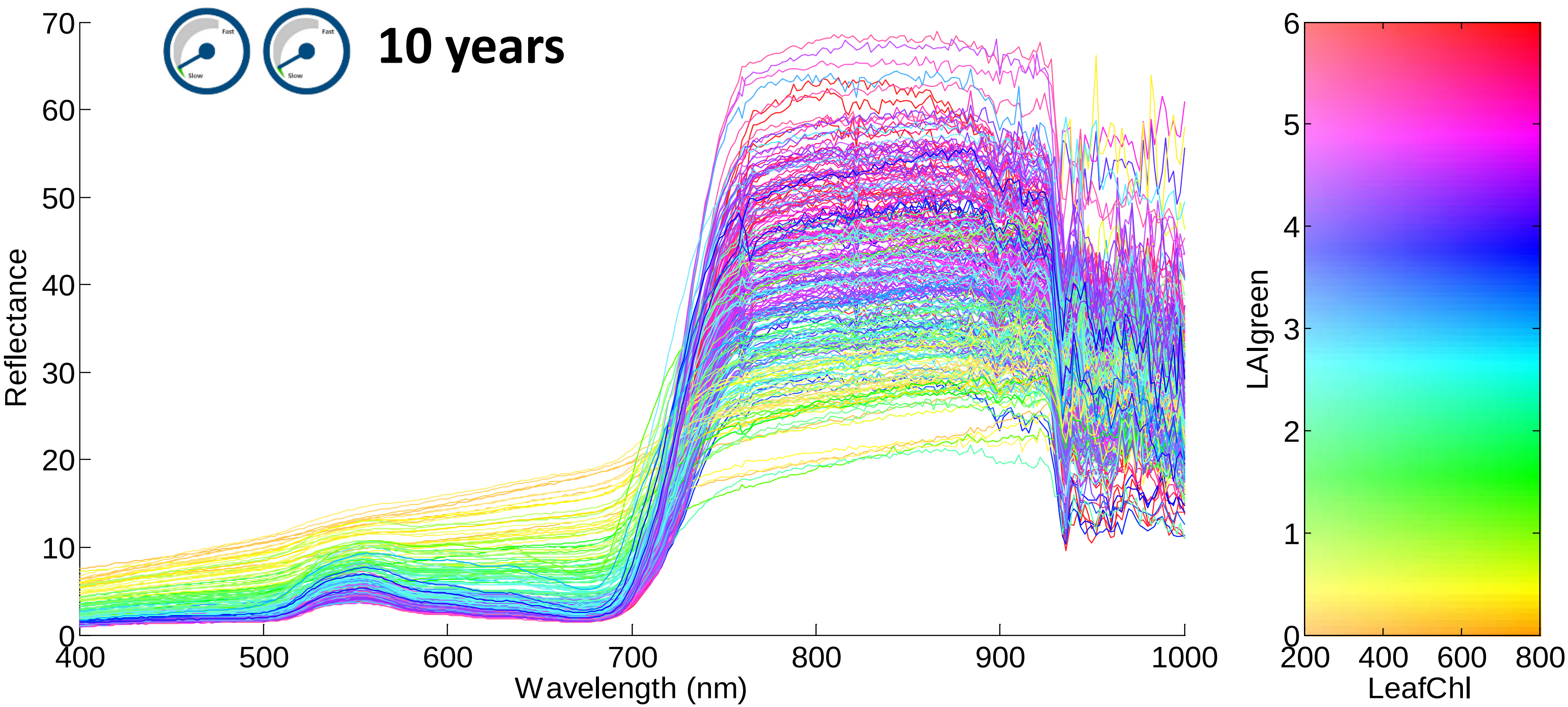

***Using MODTRAN would take <u>more than a month</u>.***

# Field data

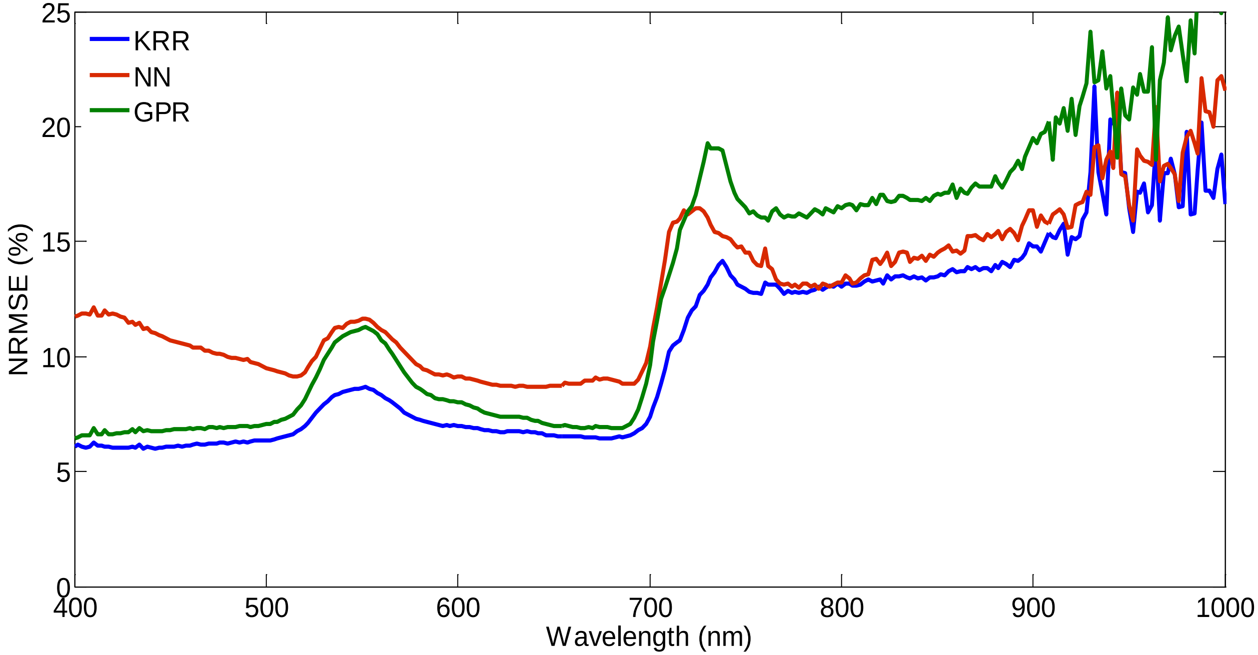
# UNL dataset *(thanks to Anatoly ☺)*

- **>260 samples over maize and soya**
- **Multiple variables measured: Cab, gLAI, FPAR, GPP**
- **Ocean Optics: 400-1000 nm**
- **301 bands at 2 nm**

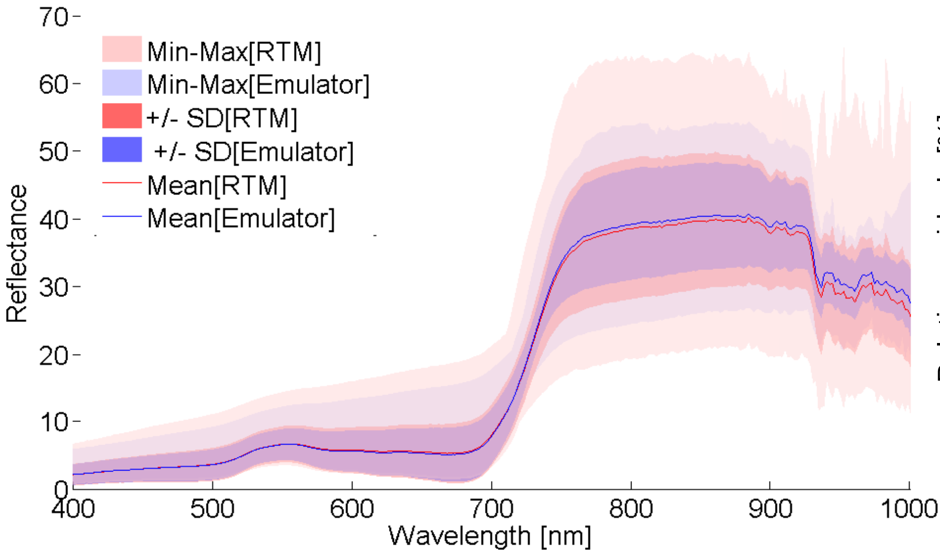
**10 years**

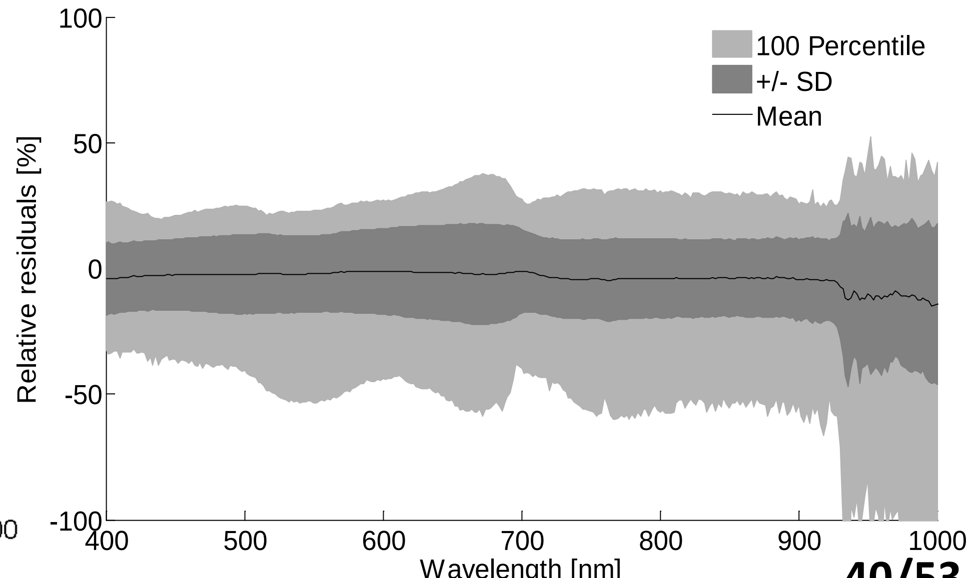


# Emulating field data (50PCA, 80/20%)



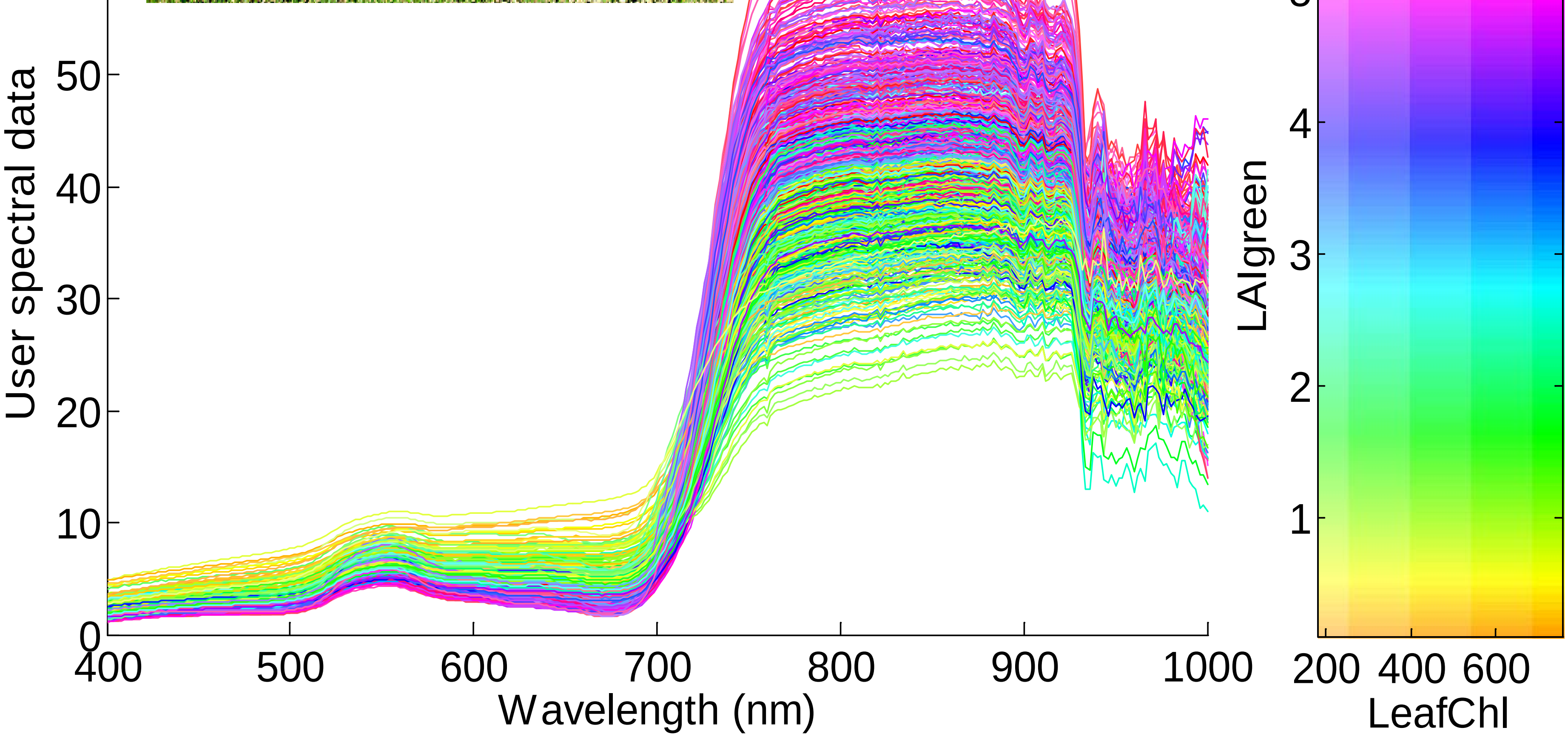

**KRR**



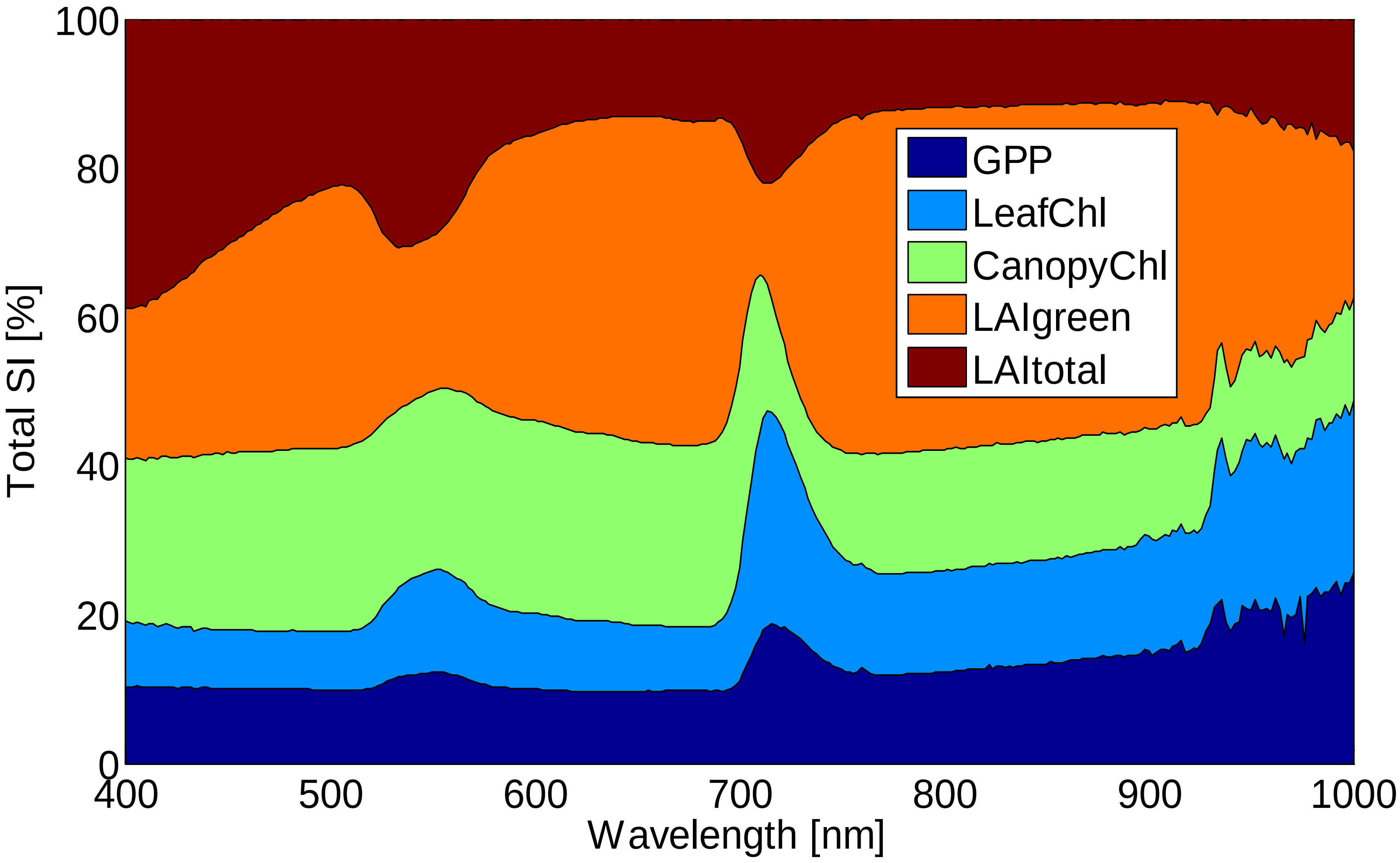
# Emulation #1000 field spectra

**25 s**



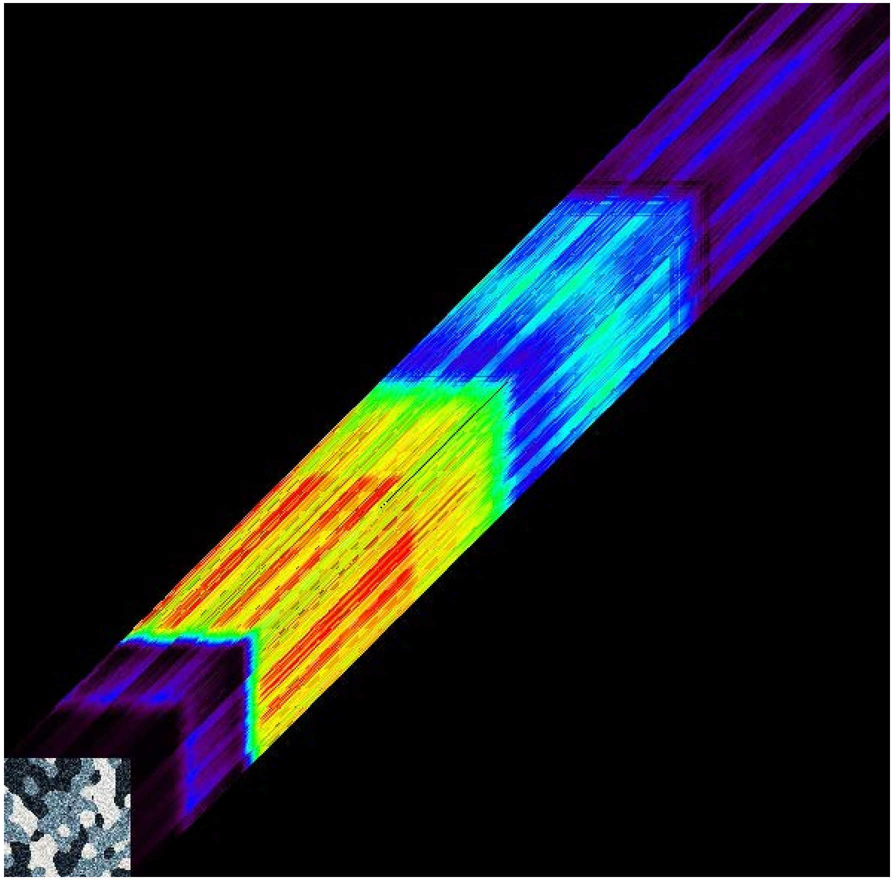
# GSA field dataset

11 s

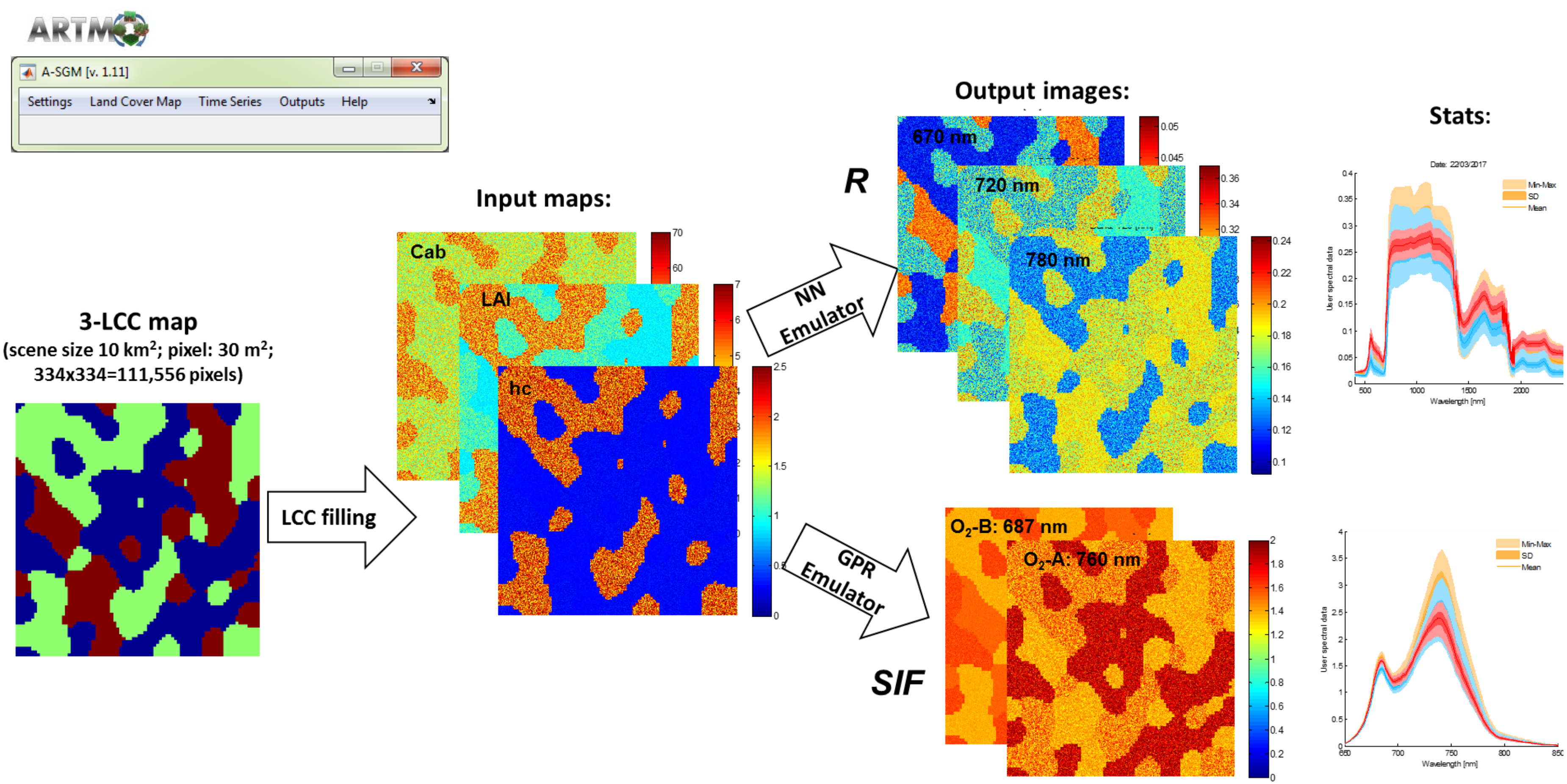



- **Peak in red edge driven by GPP, LAIgreen and LAItotal**
- **Smaller peak in PRI region**

# Scene generation



# Using best-performing *R* and *SIF* emulator for scene generation

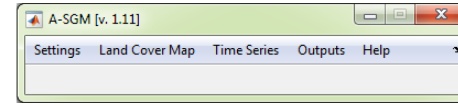



**Input scenes and hyperspectral output cubes generated by emulators (334 X334):**

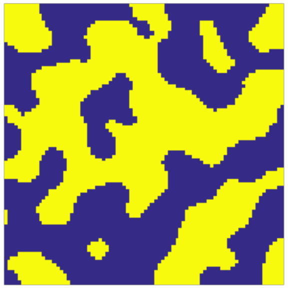
- ***R*** **by NN emulator: <u>24 min</u>**
- ***SIF*** **by GPR emulator: <u>69 min</u>**

**(SCOPE: 5 days)**

# Emulation time series scene generation



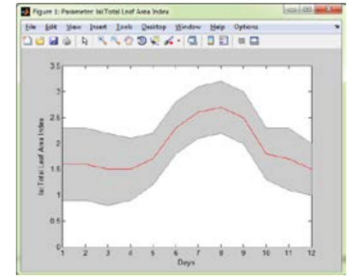

**A randomly generated 2-class image**
**(scene size 10 km²; pixel: 30 m²; 334x334=111,556 pixels)**




**Ranging variables:**

- ✓ LAI
- ✓ Cab
- ✓ Vcmo
- ✓ Vegetation height

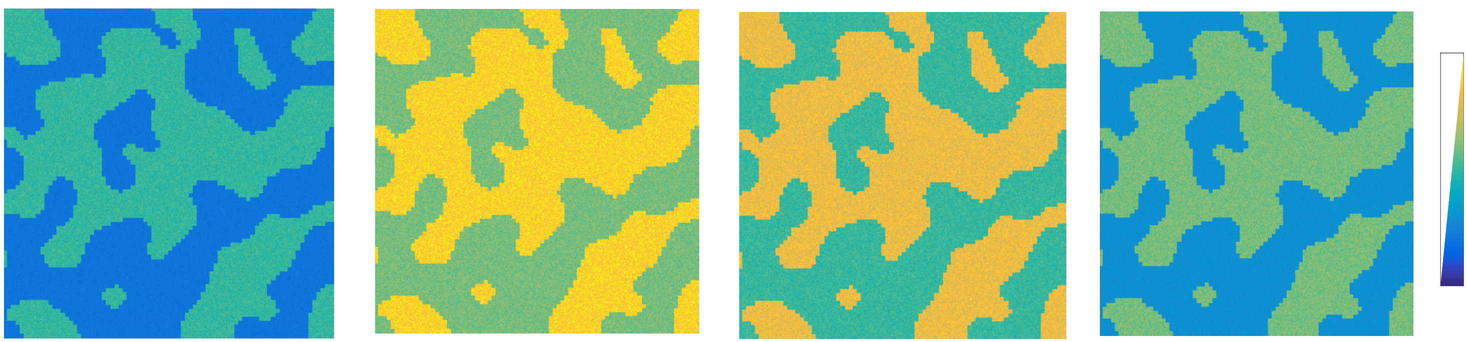
**Input temporal profile: e.g. LAI**



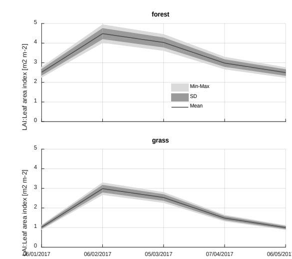

## Input maps:

Time 1 Time 2 Time 3 Time 4

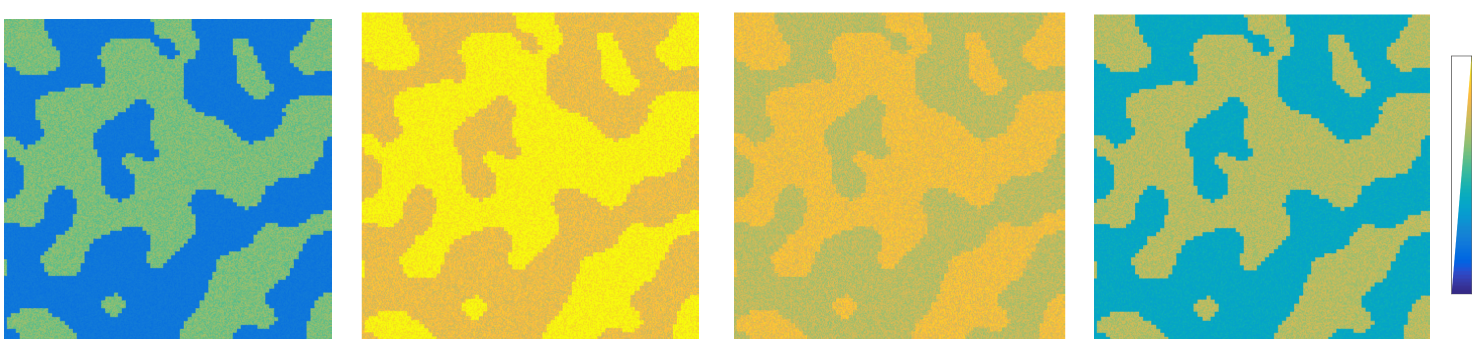
**LAI**

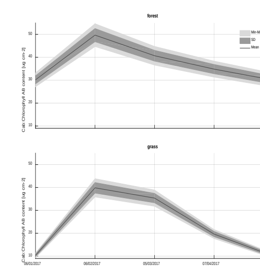


*Temporal profiles*




**Cab**

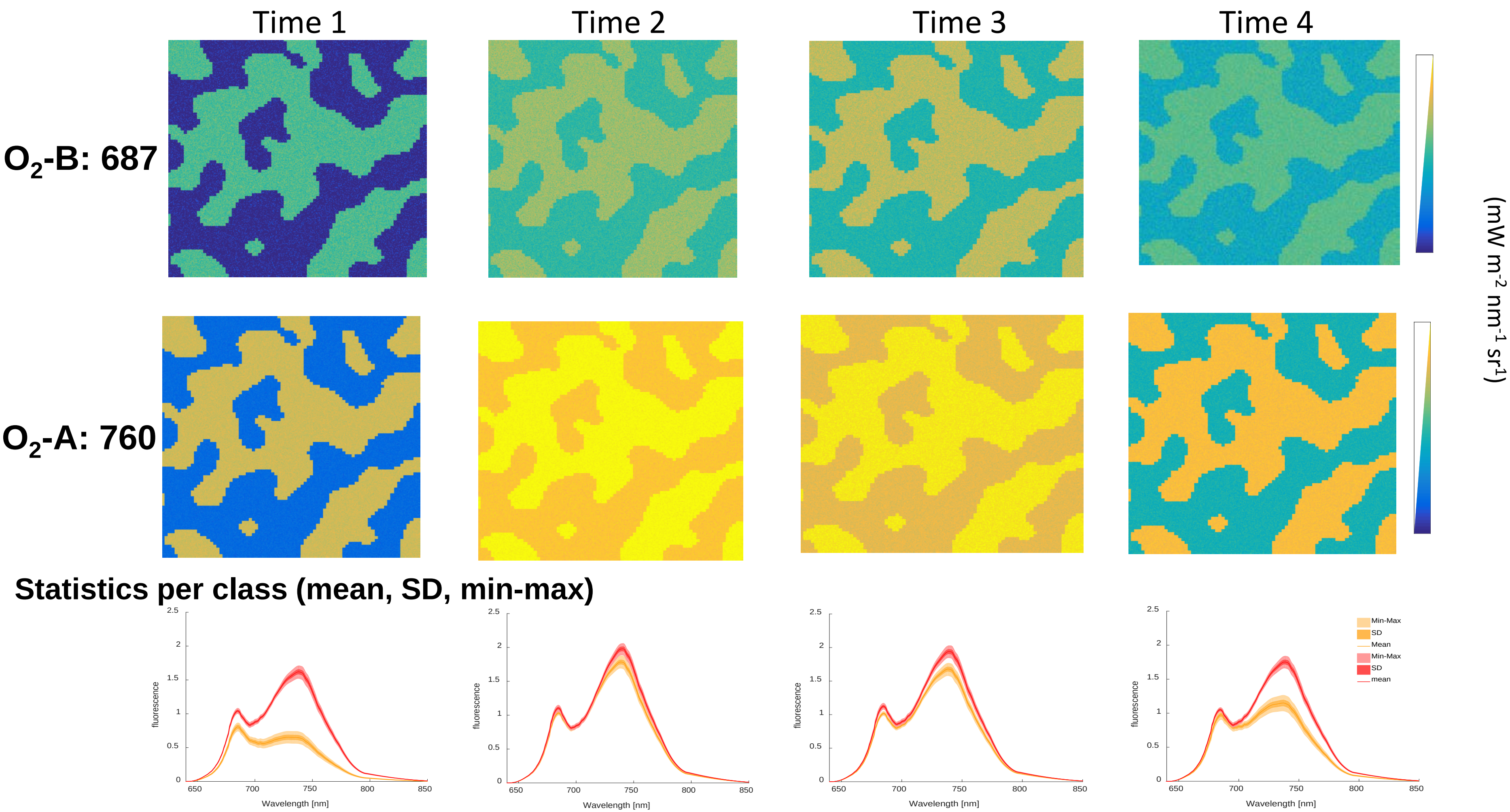




# *SIF* outputs:

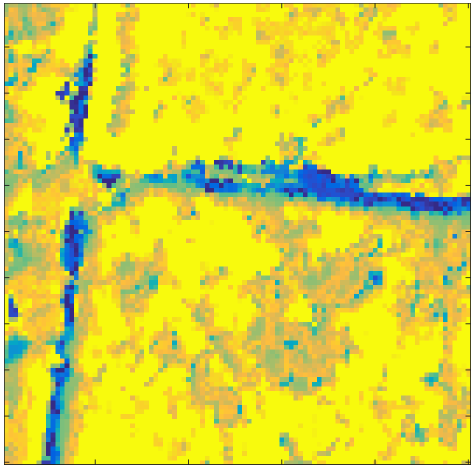
GPR: < 3 hours
KRR: < 4 min.

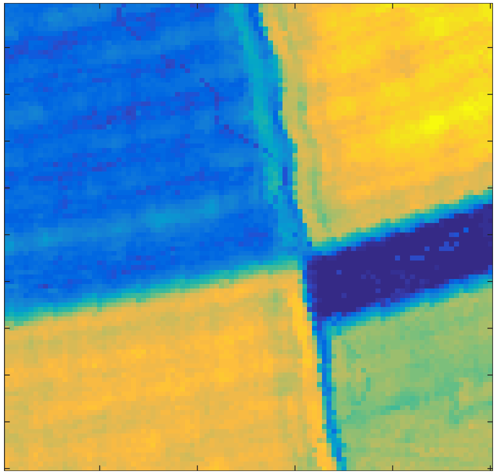
(SCOPE: 5 x 4 days)




- ✓ ***Generation of time series in an instant. Time for real image generation.***
- ✓ ***Coupling with MODTRAN emulator would enable TOA generation.***

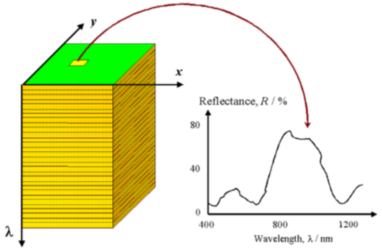
# Numerical inversion: spectral fitting



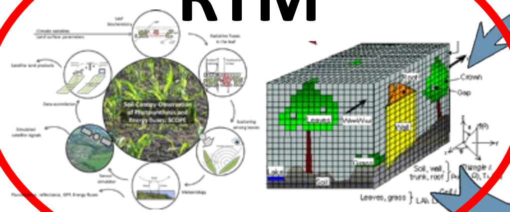


# Emulators into numerical inversion

**Image**

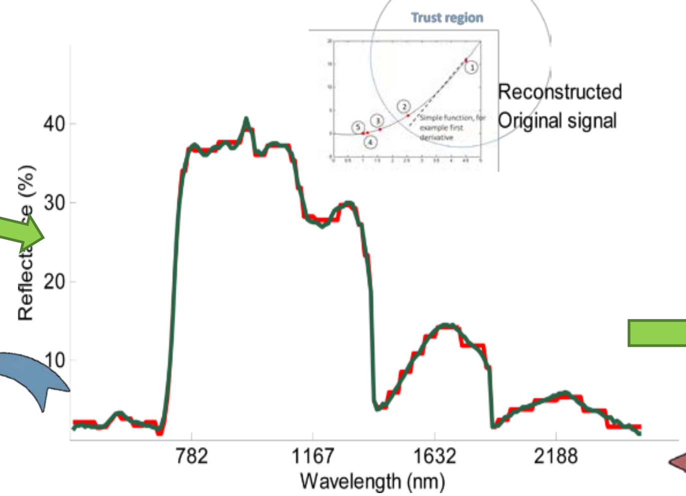



**RTM**

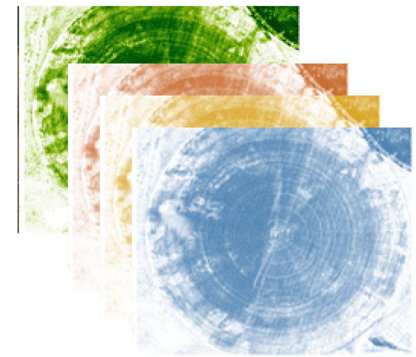


**Spectral fitting:**

Minimization algorithm: lsqnonlin

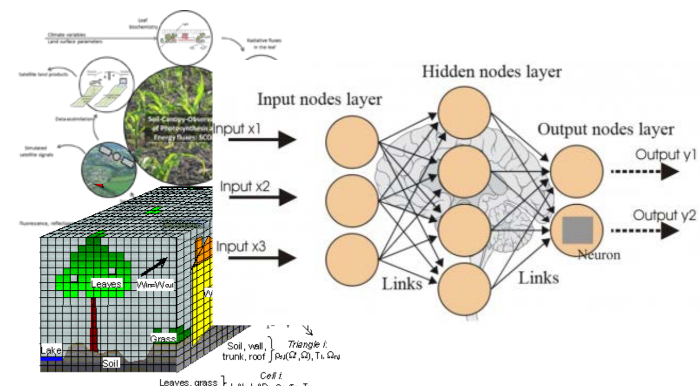



**Output maps of RTM variables**



*Per-pixel RTM iterations: very slow method, inapplicable to advanced RTMs.*


**Emulation of an RTM**

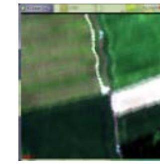



**Forest** **Agriculture**

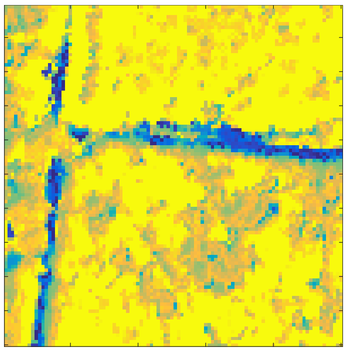
**DART** KRR emulator applied to HyPlant DUAL (450-2500 nm)


< 1 h

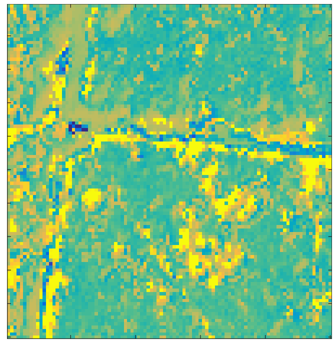
**SCOPE** KRR emulator applied HyPlant DUAL (bare soil spectra added)



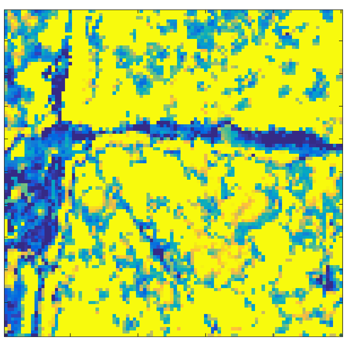
**CC**



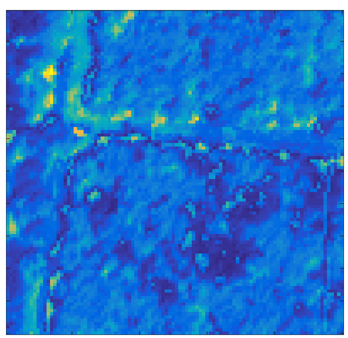
**LCC**



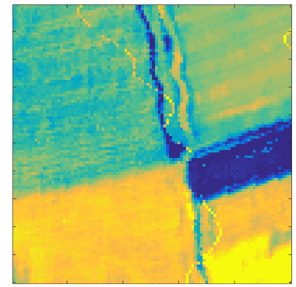
**LAI**



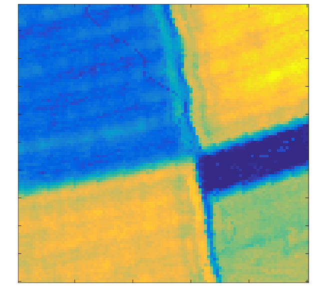
**RMSE**



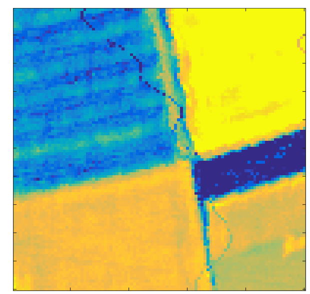
**APAR**



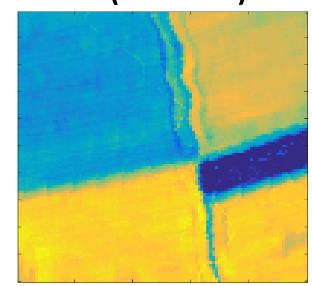
**LAI**



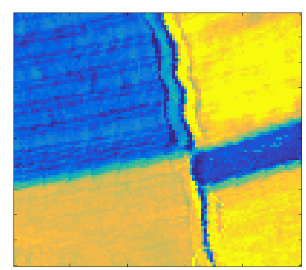
**fAPAR**



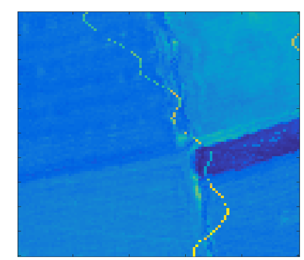
**CCC (LCC x LAI)**



**CWC (Cw x LAI)**

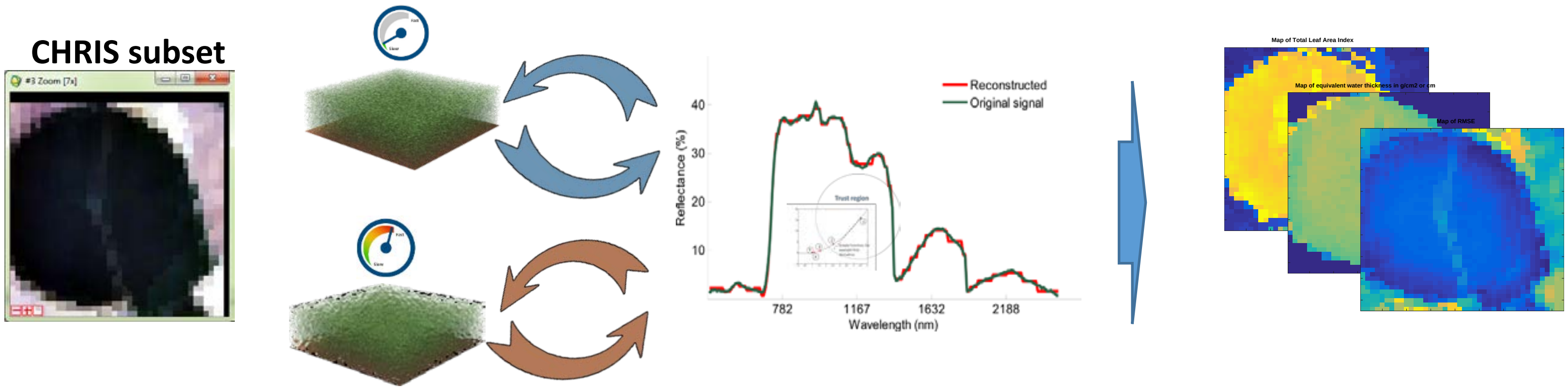


**RMSE**

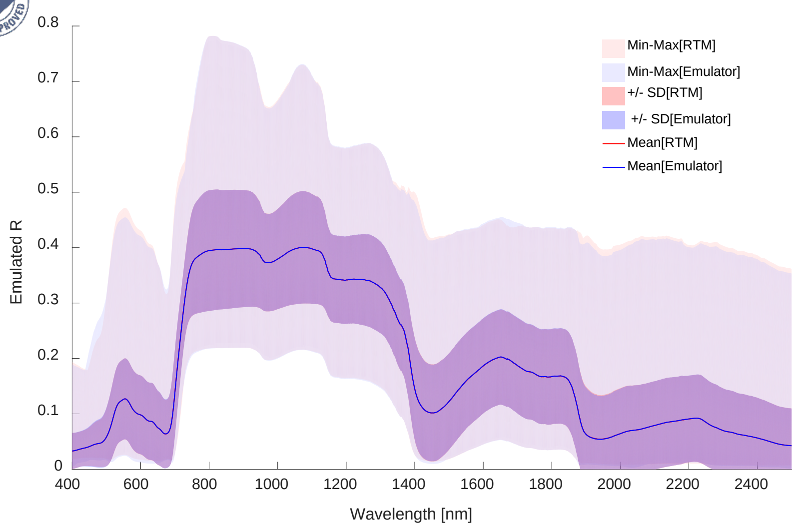


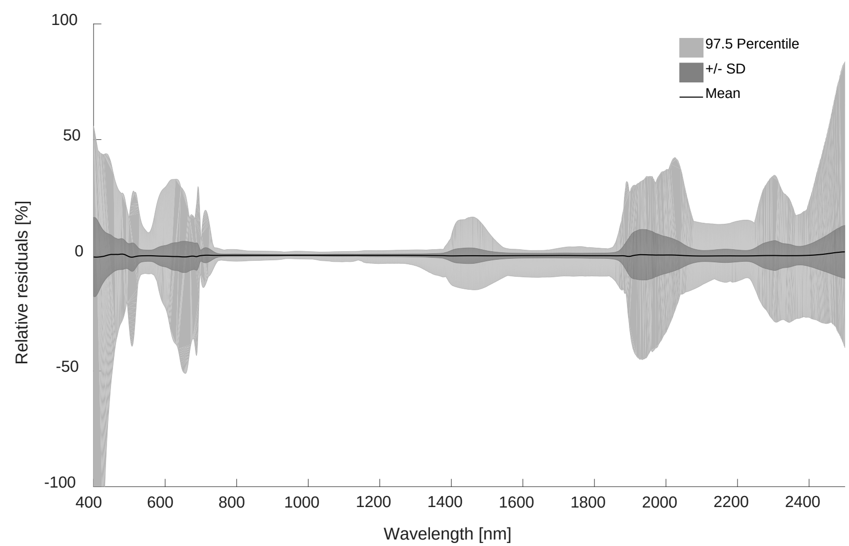
**Retrieval quality depends on : (1) emulator, (2) number and type of included variables.**

# Comparison PROSAIL vs emulator




## Validation NN emulator (1000#, 20 PCA):



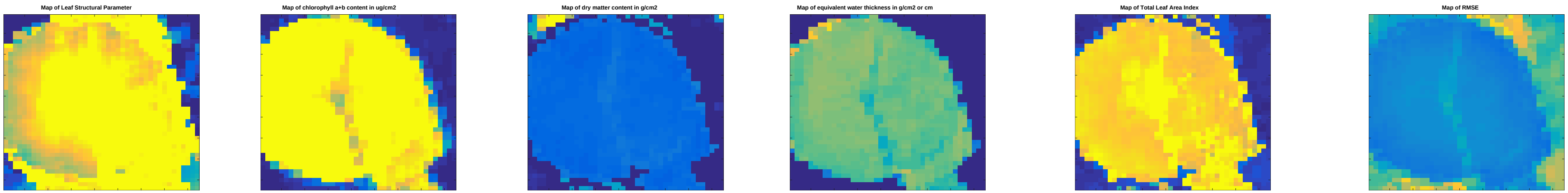



# Numerical inversion comparison

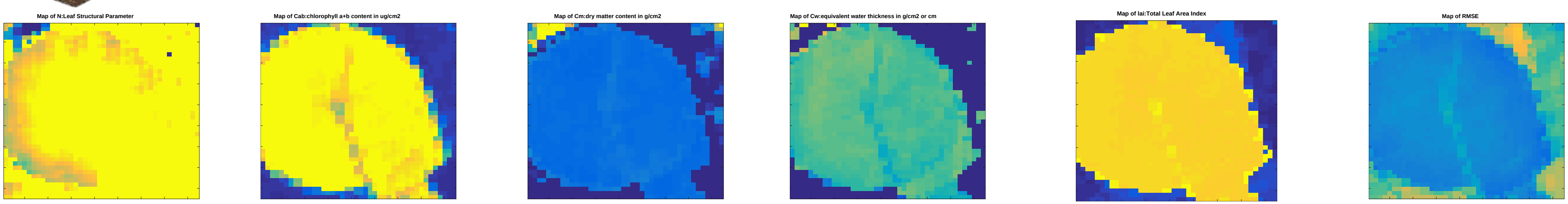
X 3.1

## PROSAIL: 66 min

N | Cab | Cdm | Cw | LAI | RMSE




## NN emulator: 21 min




- **Gain in speed while about same outputs maps delivered**.
- Some outputs emulator failed: further fine-tuning required.
- PROSAIL is already very fast, **the gain in processing speed will be more substantial when using emulators of advanced RTMs**.
- **Using emulator takes hardly memory space** (no longer need for heavy LUTs).

# Conclusions

- Thanks to machine learning, **RTMs can be emulated**.
- **Trade-off:** enormous gain in processing speed at expense of some loss in accuracy.

## Advantages emulation:

✓ **Any physical model can be emulated**

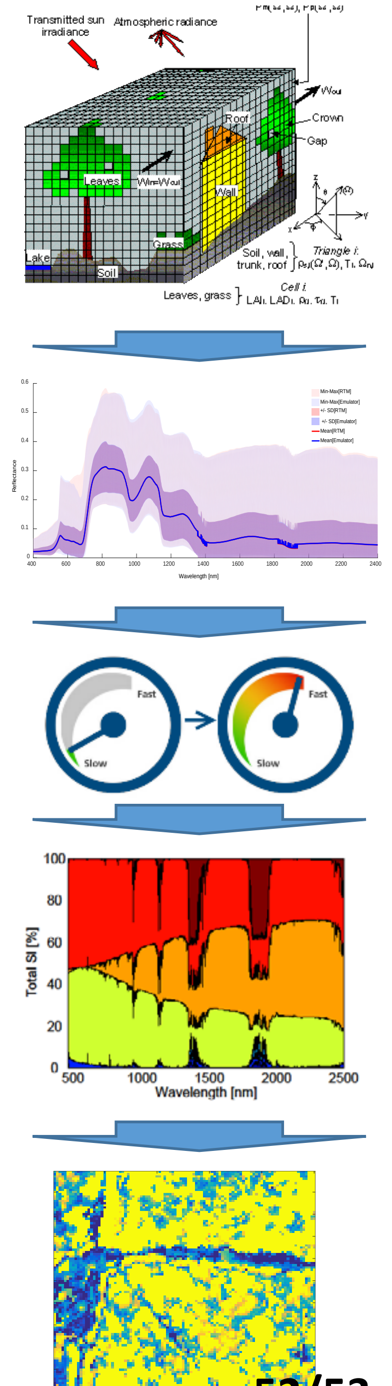
✓ **Fast generation of large simulated datasets (LUTs)**

✓ **Fast emulation of field data**

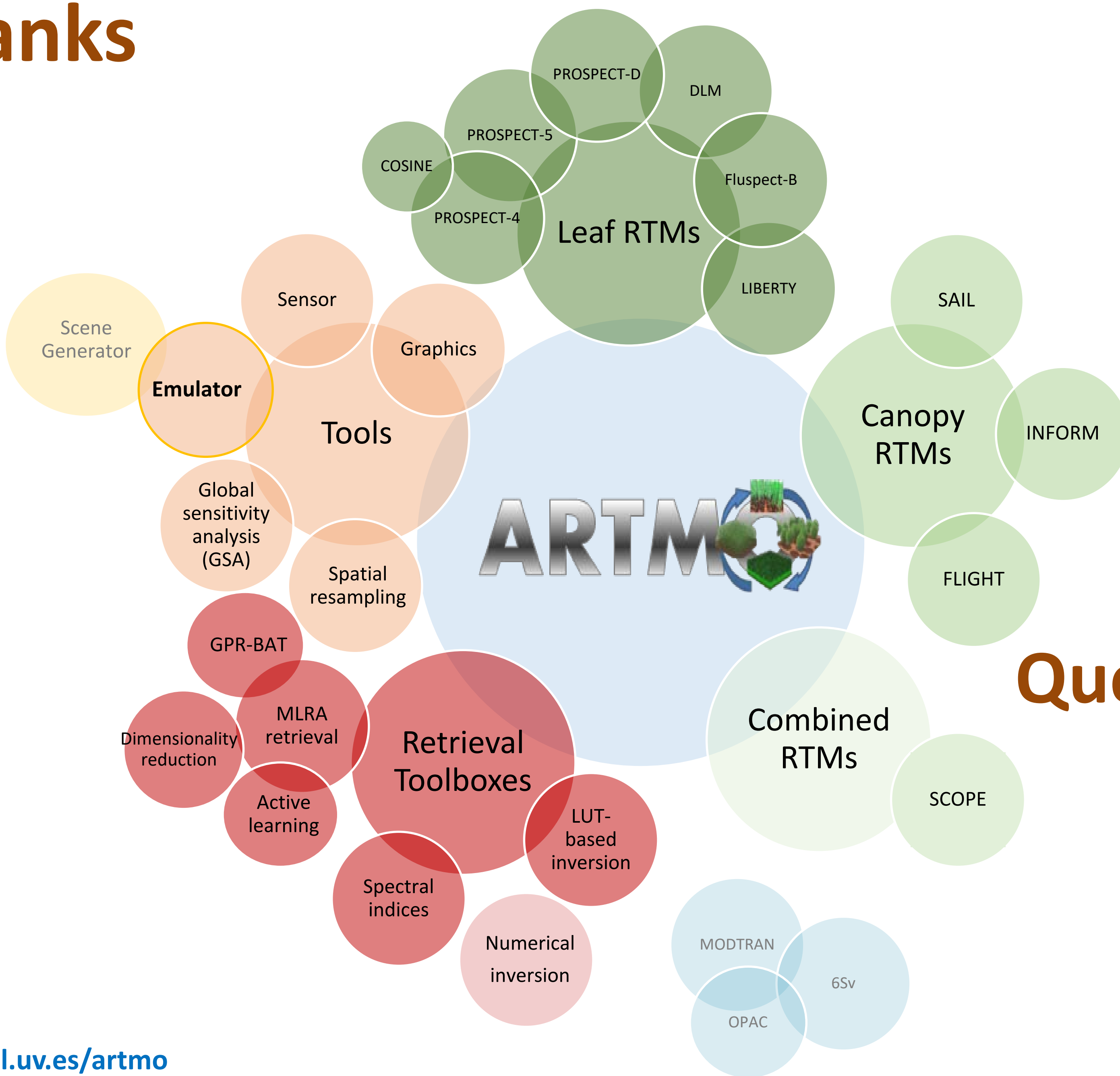
✓ **Fast global sensitivity analysis**

✓ **Fast synthetic scene generation**

✓ **It makes numerical inversion again an attractive method ☺**



# Thanks




# Questions?

ipl.uv.es/artmo
isp.uv.es/software