Global scale land cover mapping using free and open-source geospatial software

Dainius Masiliūnas[1], *Nandin-Erdene Tsendbazar*[1], *Martin Herold*[1], *Myroslava Lesiv*[2], *Marcel Buchhorn*[3], *Bruno Smets*[3], *Niels Souverijns*[3], *Jan Verbesselt*[1]

1) WAGENINGEN UNIVERSITY & RESEARCH 2) IIASA 3) vito

Content

- Copernicus Global Land Operations – Land Cover project
- Global land cover fraction mapping
- Land cover change detection
- BFAST – breaks for additive season and trend
- OpenEO project

Land cover monitoring and updating

- Land cover maps: what we can see on the ground (trees, grass, water, urban, …)
- Key variable for several UN Sustainable Development Goals
    - Governments need to set policy
    - Land owners want to know what is present
    - LC change allows monitoring the current situation and predicting future change
- We can do it globally thanks to satellite imagery time series
- Updating is essential for effective monitoring

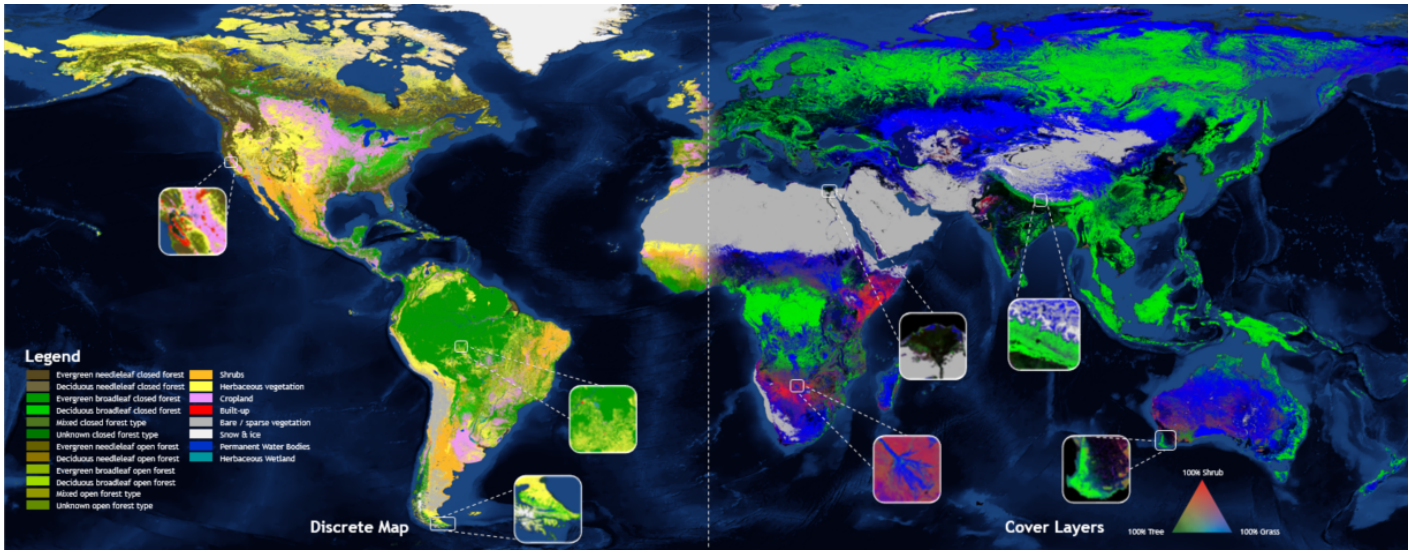



WAGENINGEN
UNIVERSITY & RESEARCH

Copernicus Global Land Services

- Operational services for global land monitoring
- CGLS-LC100
    - We have: 21 discrete classes and 10 continuous covers (class fractions) globally
    - From 2015 updated yearly, as an operational service

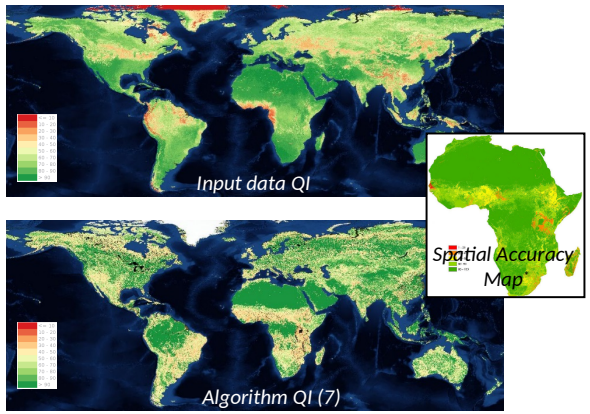
- My contribution:
    1) Land cover fraction mapping methods
    2) Change detection for yearly updating



land.copernicus.eu/global/lcviewer

Copernicus Global Land Services Land Cover Collection 3

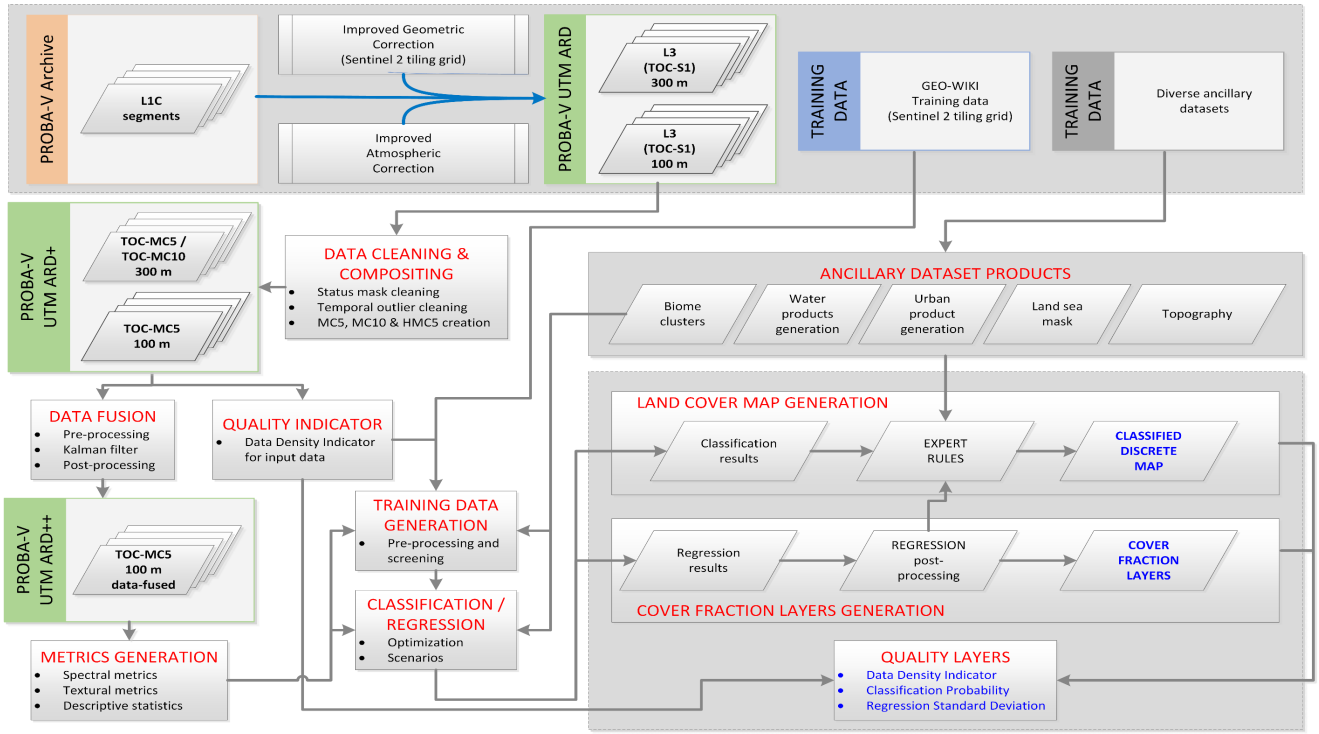
- To be released in a few days! Keep track of the release progress:
  - https://twitter.com/VITO_RS_
  - https://twitter.com/CopernicusLand
  - https://twitter.com/CopernicusEU
- Includes yearly updates for the whole globe and quality indicators for each pixel!



Quality Indicators

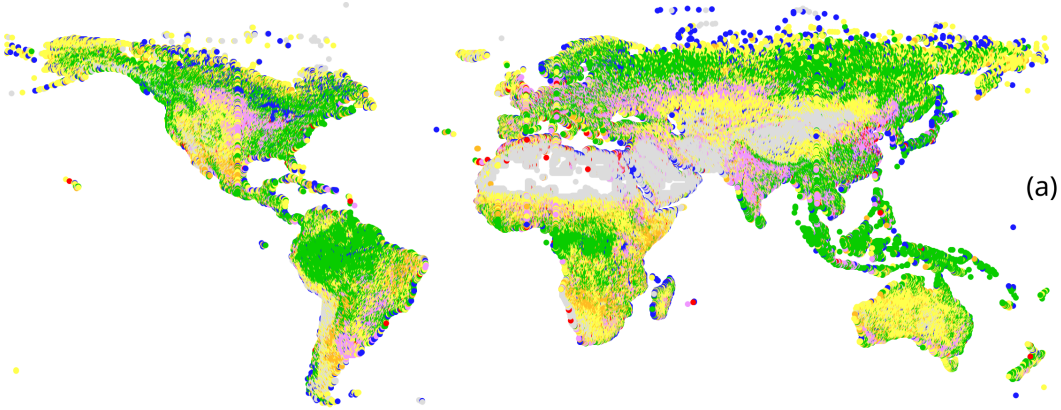
(*) example over Africa, global maps under release test

CGLS-LC100 processing chain

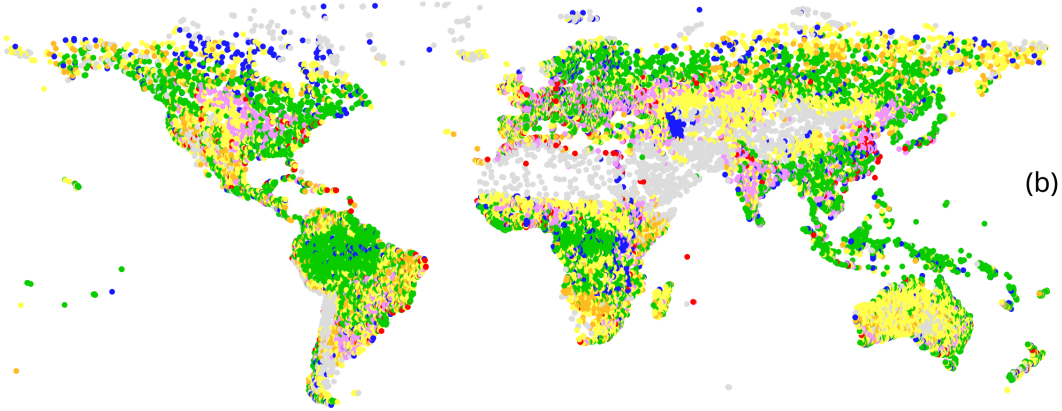


CGLS-LC100 reference data

Training data: 150 405 points with fraction data for model training (IIASA)



Validation data: 21 752 points with fraction data for model validation (WUR)

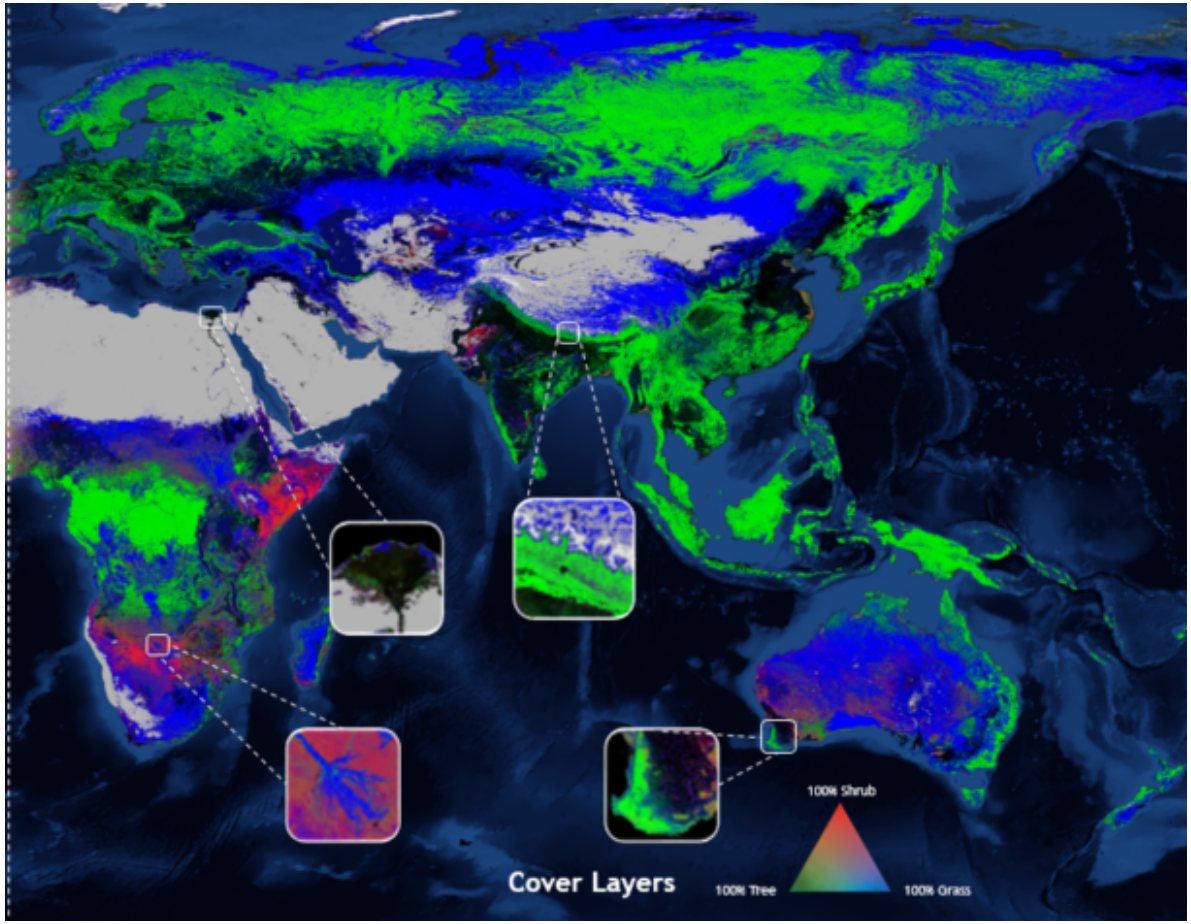


Legend

Dominant LC class

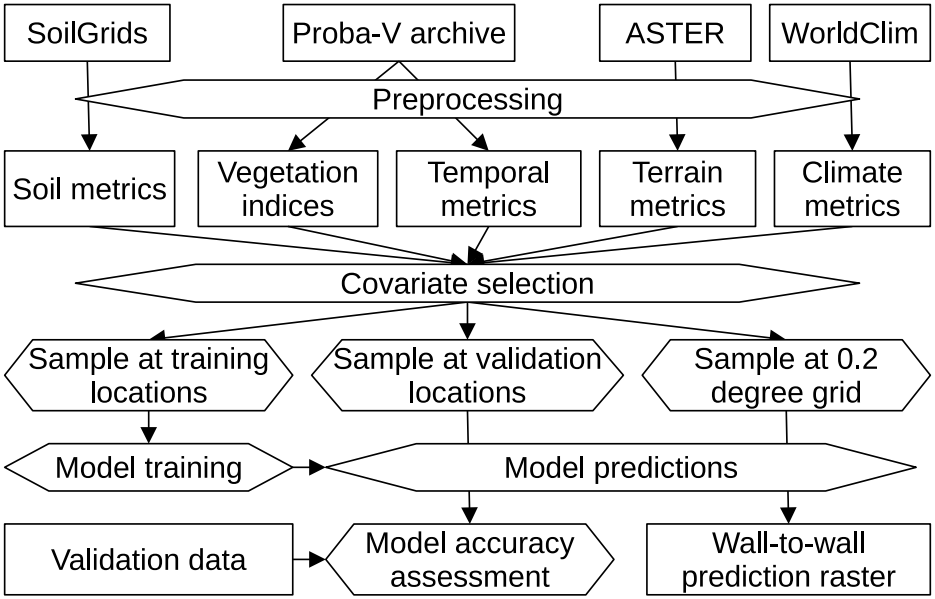
- Trees
- Shrubs
- Grassland
- Lichen and moss
- Wetland (herbaceous)
- Crops
- Urban/built-up
- Bare
- Snow and ice
- Water
- Not sure

PhD topics: 1) Global land cover fraction mapping



Objectives

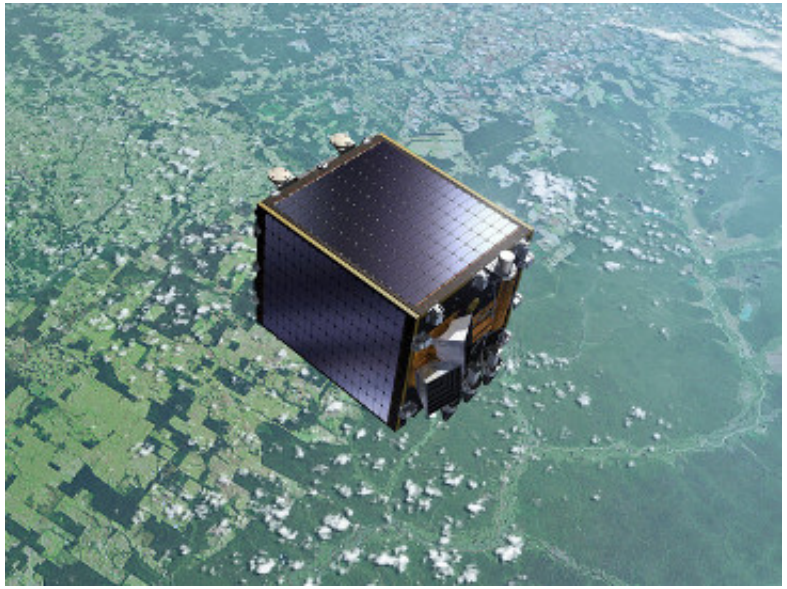
- Compare machine learning models for global land cover fraction mapping
- Tune models to account for zero inflation inherent in fraction mapping
- Compare covariate importance
- Make a global map and check its errors



67% herbaceous
33% trees
0% water
0% shrubs
0% built-up
0% crops
...

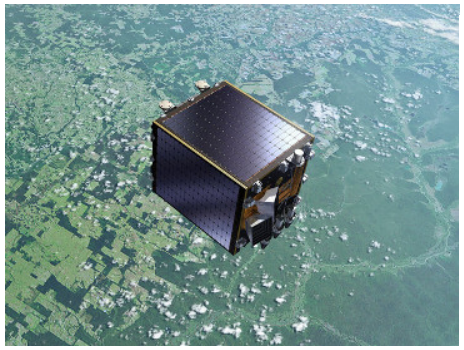
Land cover fraction mapping: model input

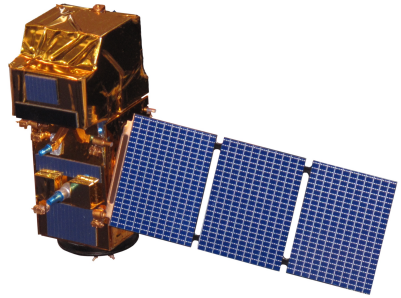
- Proba-V satellite archive (2014-today, 386 images), 4 bands: green, red, NIR, SWIR
- 5 vegetation indices: NDVI, EVI, NDMI, NIRv, OSAVI
- Over 300 covariates (all global!):
    - Spectral (TS, composites, etc.)
    - Terrain (slope, roughness, etc.)
    - Climate (temperature/month, etc.)
    - Soil (SOC, bulk density, etc.)

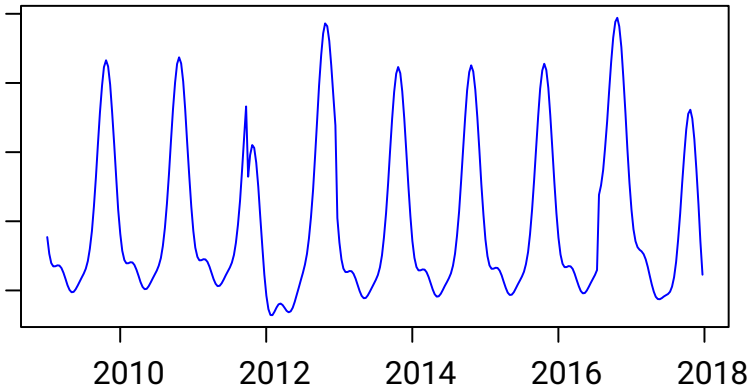


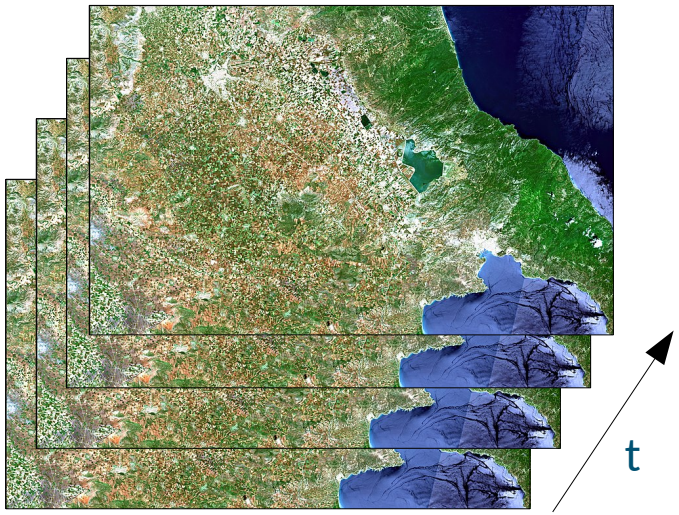
Big data challenges

- Global datasets
- Importance of time series
- 100 m to 20 m upscaling
- Multidimensionality: X*Y*Time*(Bands+VIs+covariates)
- CSVs don’t scale well









Terrascope

- VM (4 cores, 8 GiB RAM, 1 TiB HDD, CentOS 7.4)
- Direct (NFS) access to L3 TOC composite data (radiometry + NDVI), including ARD, + ASTER and MODIS VIs since 2009
- Access to Apache Spark cluster
- Some data (climate, soil) needs to be downloaded
- Takes a while to extract point time series: reading 386 files for each point (461892) takes over a month (using GDAL via Bash, similar to using GDAL via Python)

CentOS

GDAL

APACHE Spark™

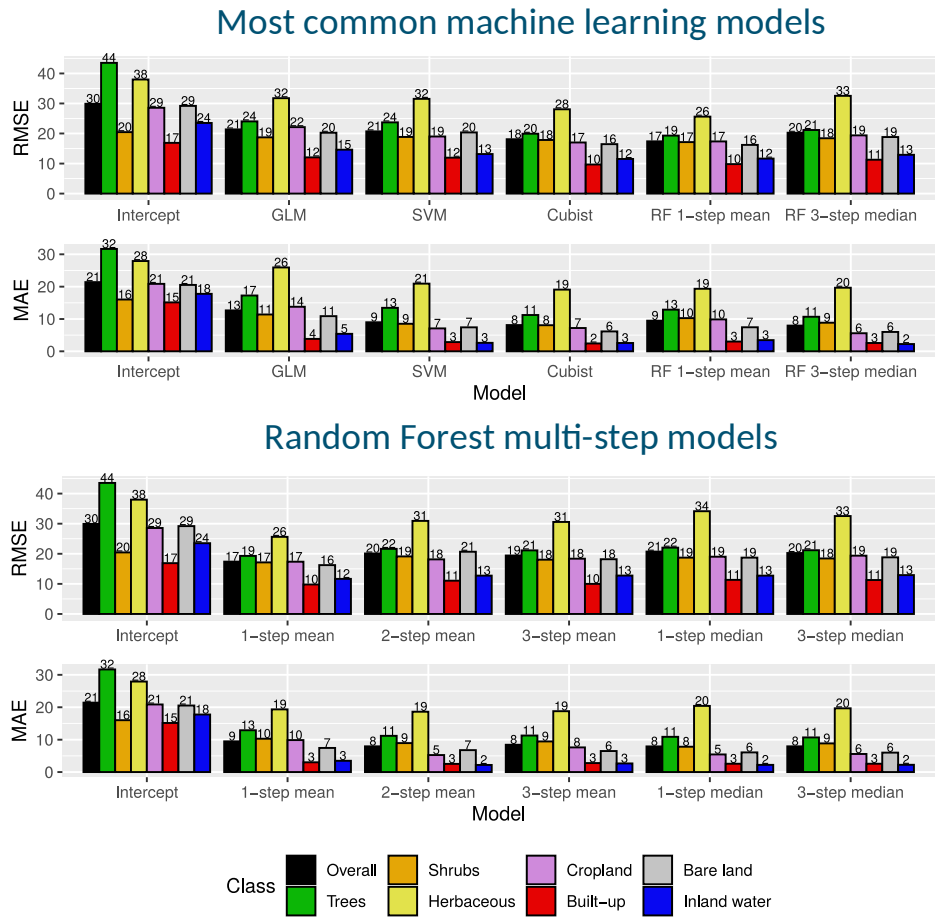
WAGENINGEN
UNIVERSITY & RESEARCH

Land cover fraction mapping: algorithms

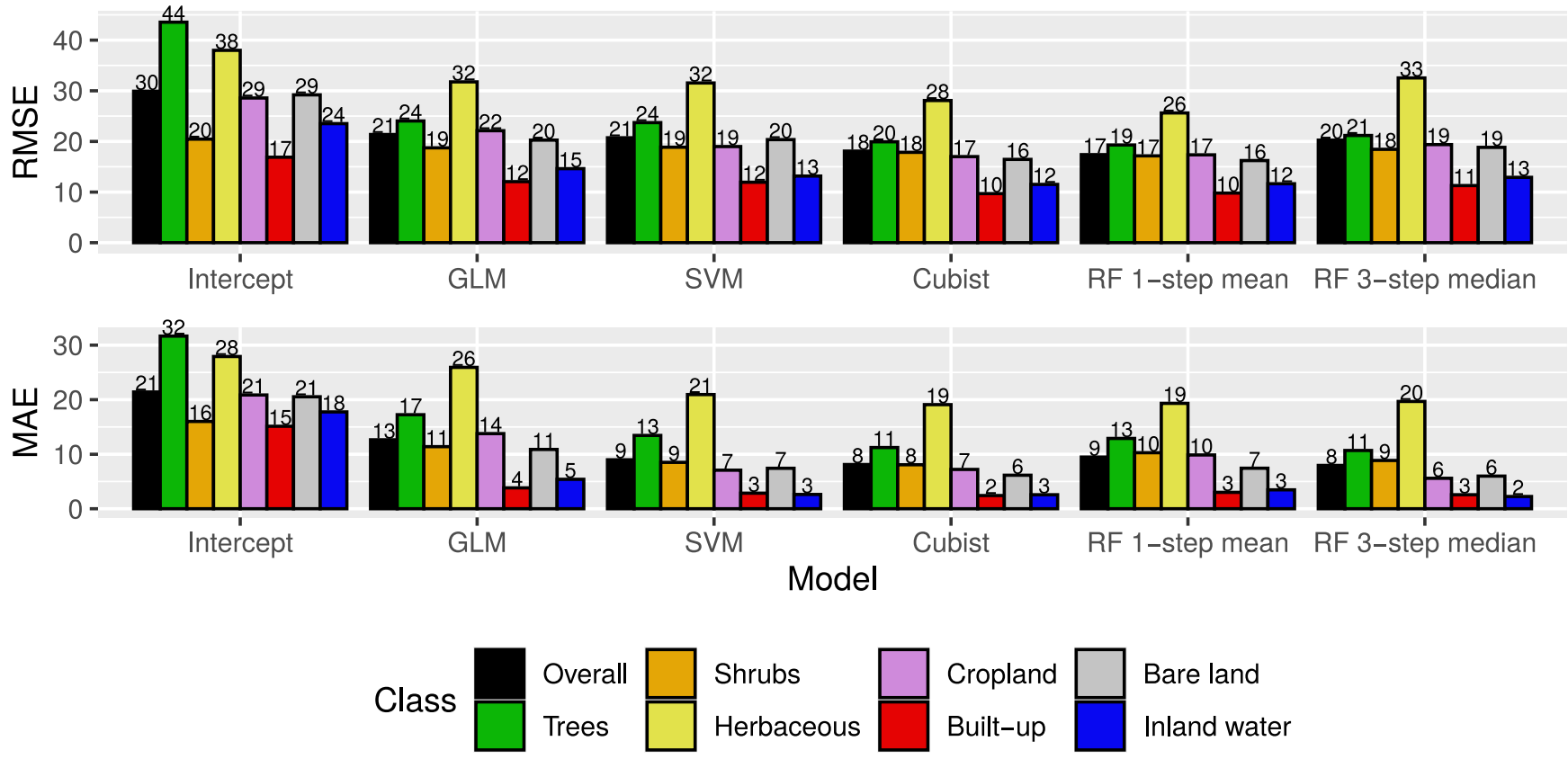
- Machine learning regression algorithms in R: general linear model (stats), fuzzy nearest centroid (GSIF), logistic regression (nnet), lasso regression (glmnet), Partial Least Squares regression (plsr), Random Forest (ranger), artificial neural networks (keras), Cubist (Cubist), support vector machines (liquidSVM)
- Keras for neural networks:
    - Binding to Python (via reticulate)
    - Backend: Tensorflow
    - Accelerator: NVIDIA GTX 660
- Tried also MESMA, SuperLearner and MultivariateRandomForest but didn't work

Land cover fraction mapping: model performance

- Random Forest is the most accurate: 17.3 RMSE, 9.4 MAE, 0.66 NSE, 67±4% OA
- Can be optimised (2-step: separate model for zeroes, separate for non-zeroes) to decrease MAE at the cost of RMSE for middle predictions
- Median voting: much better prediction of 0/100%, but more often completely misses
- 3-step + median combination is best for MAE: 20.2 RMSE, 7.9 MAE, 0.54 NSE, 72±2% OA

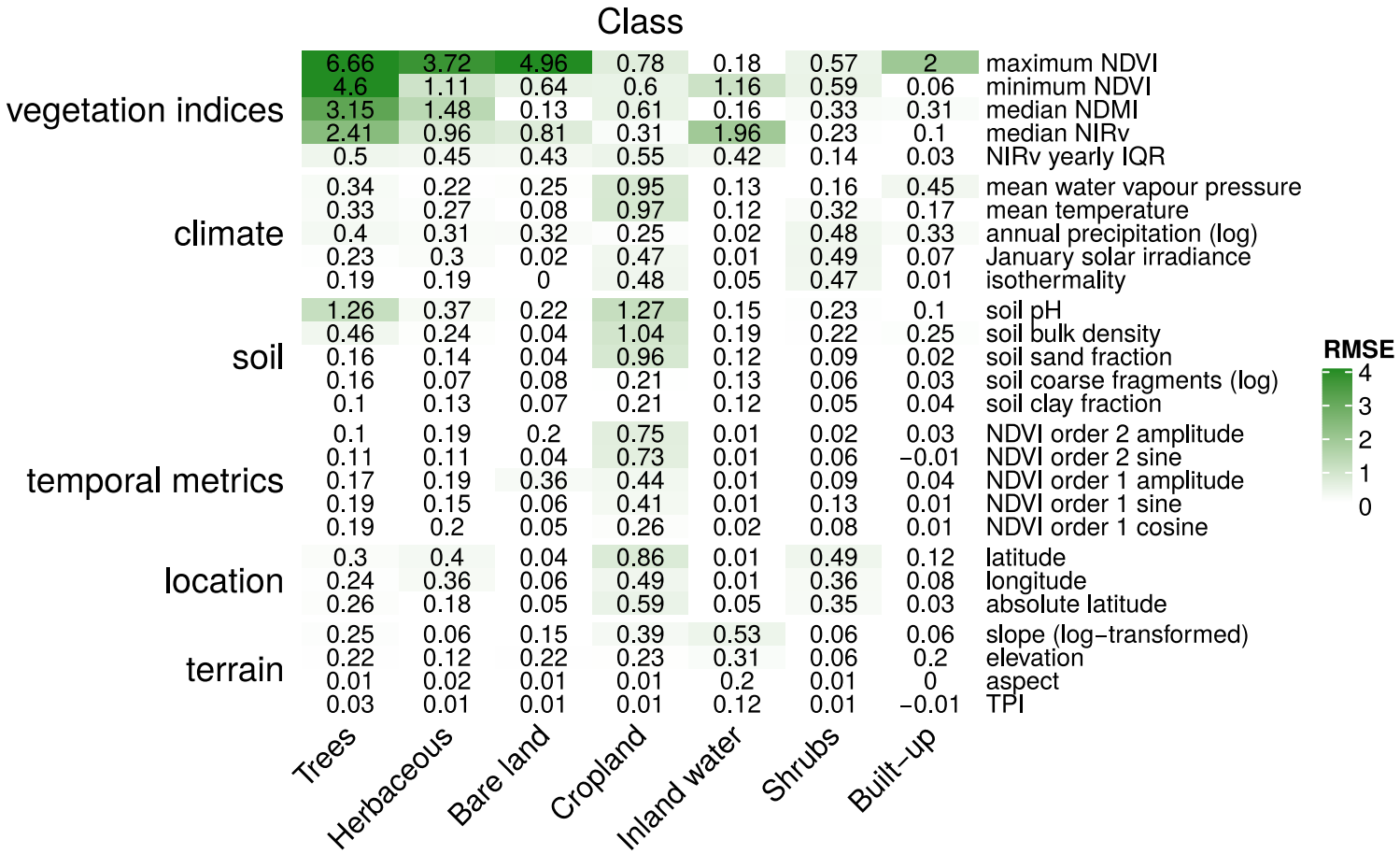


Errors per class

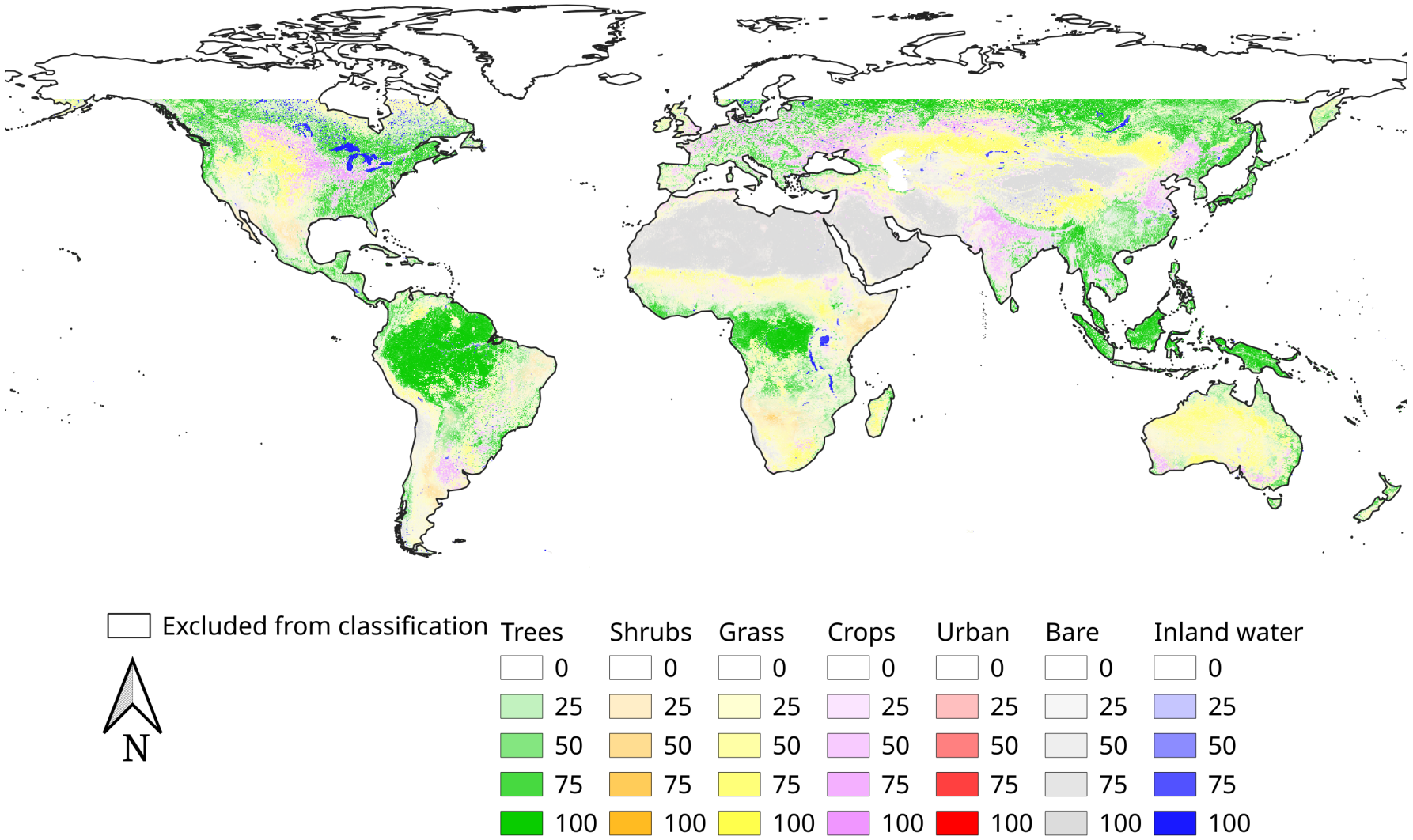


RF 1-step covariate permutation importance

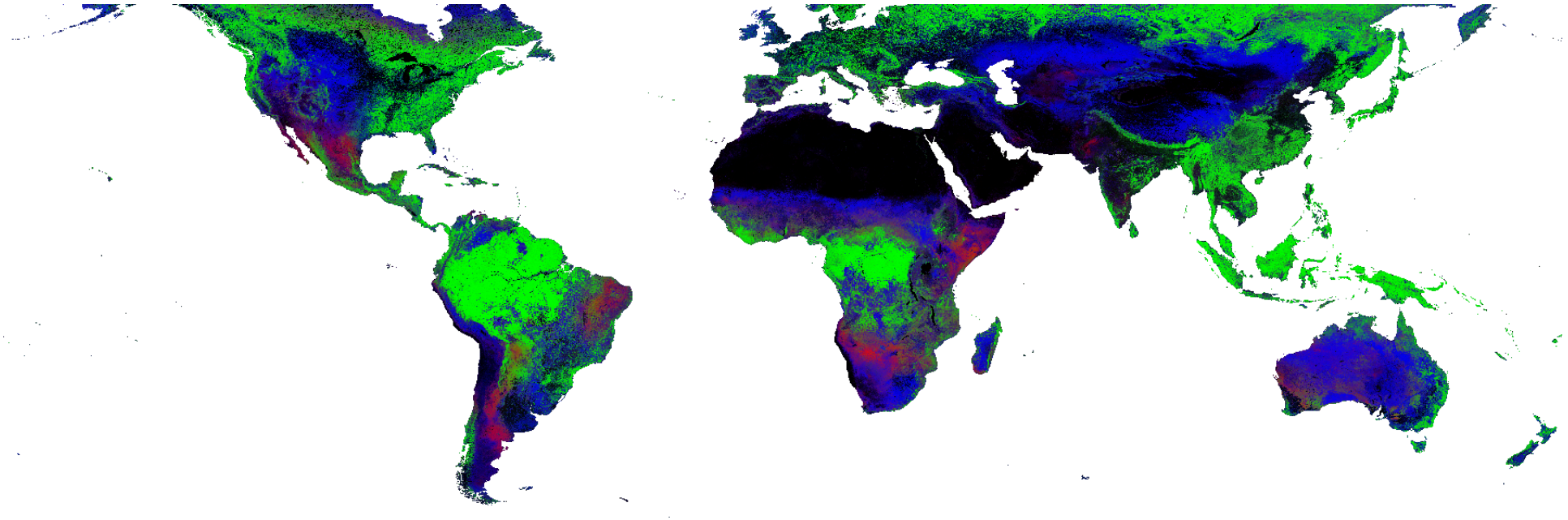
Increase in RMSE after permuting all covariates in the group



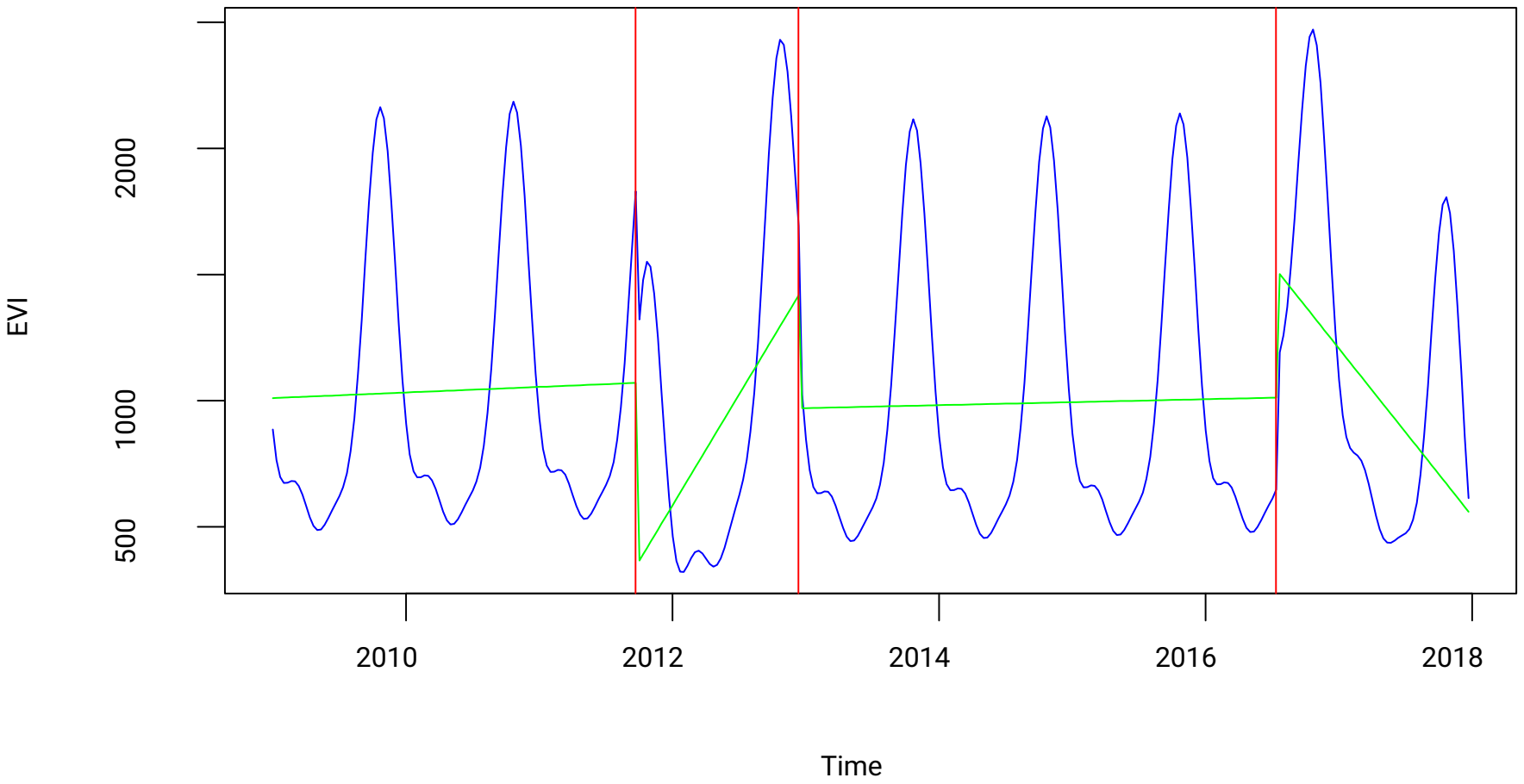
Wall-to-wall map



Shrubs, trees, grass as RGB layers



PhD topics: 2) Land cover change detection for map updating



WAGENINGEN
UNIVERSITY & RESEARCH

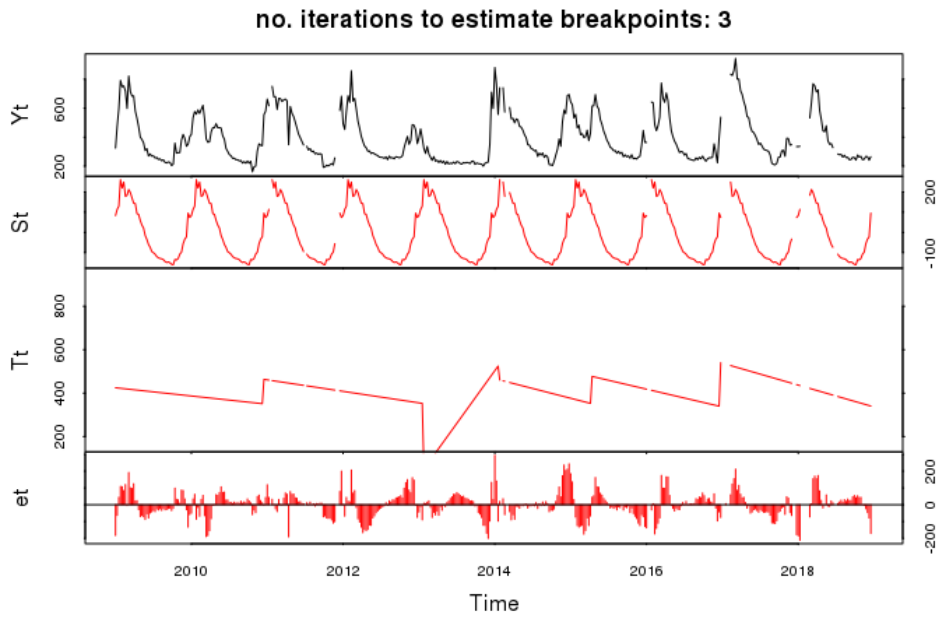
Land cover change detection for updating

- Reusing the same model for the next year leads to too many spurious changes
- Expert rules: which transitions are possible/likely
- Use time series break detection to constrain changed pixels
- But which break detection algorithm and VI?

Unlikely land cover change: from urban to water

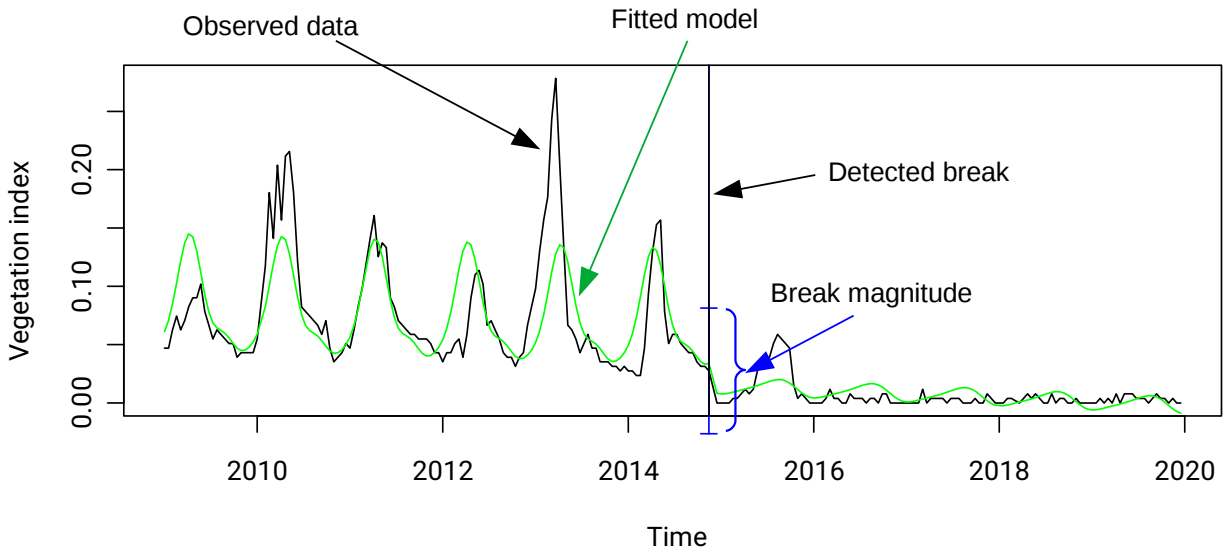
BFAST

- BFAST: Breaks For Additive Season and Trend
- Decomposition of time series into seasonal, trend and remainder components
- Components subdivided into stable segments (determined by BIC), segment divisions are breaks
- Detects all breaks, as long as it's less than h samples away from the ends of the time series (h = minimum segment size)



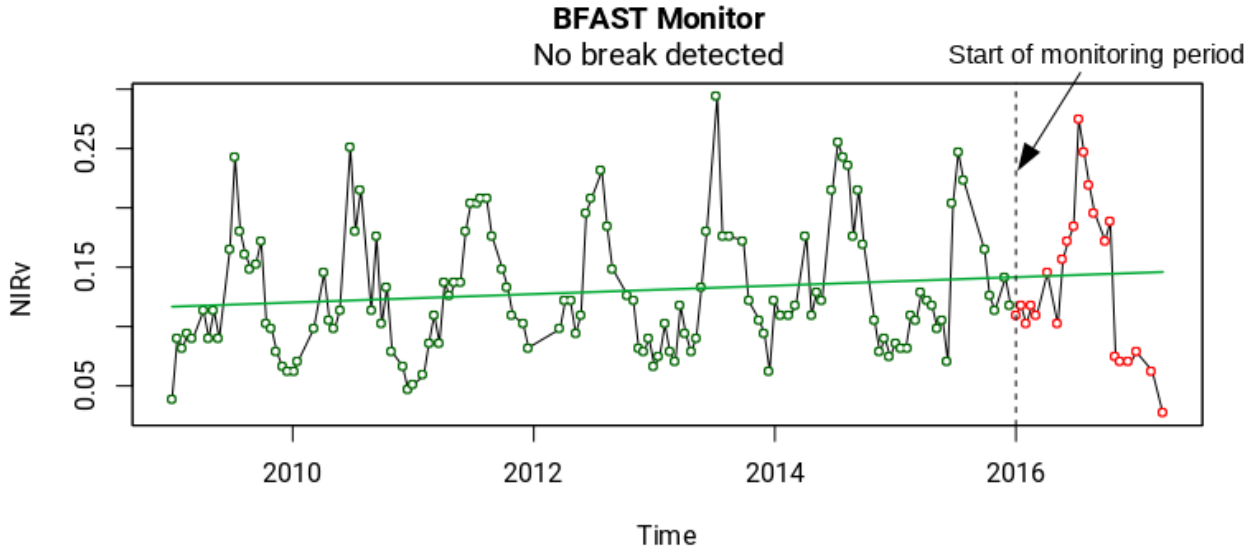
BFAST0N (BFAST Lite)

- Detecting breaks in all components at once in a single pass
- Can handle missing values
- Can use harmonics or seasonal dummies or external regressors to fit the data
- Is an order of magnitude faster than BFAST (in addition to speed improvements by Marius Appel)



BFAST Monitor

- BFAST0N has the limitation of not being able to detect breaks at the end of the time series
- BFAST Monitor is made specifically to detect changes at the end of the time series
- Used for the “NRT” map (2019, as there is not yet a whole year of data available since then)

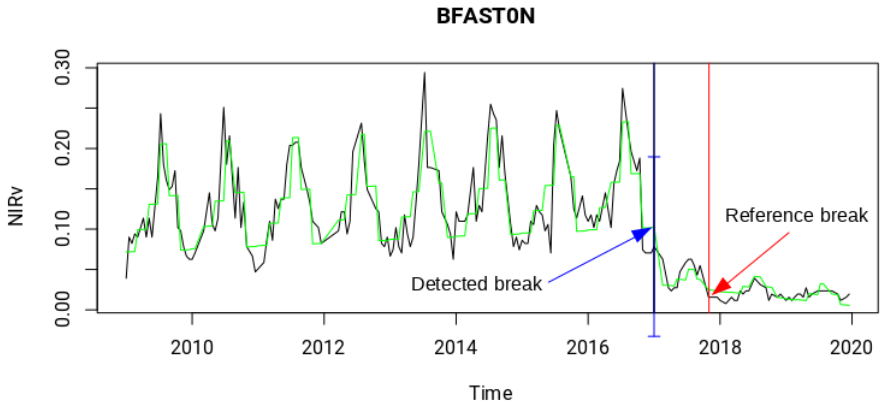


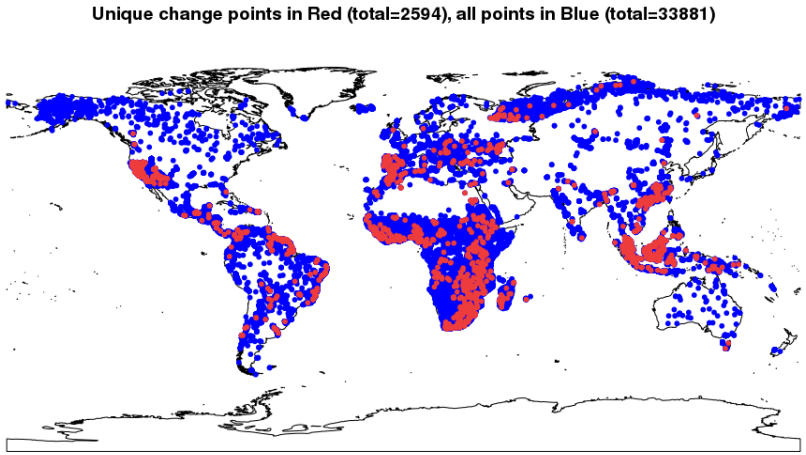
Running BFAST models on Spark

- Terrascope provides a Spark cluster with ~1200 cores
- Split each MODIS tile into ~2000 chunks using gdalbuildvrt
- BFAST is implemented in R: SparkR used to send R script to driver and executors
- Result mosaicked back to a tile locally
- Gdalbuildvrt to make a global mosaic
- In production uses rpy2: Python for I/O and R for processing data cubes

BFAST model parameter optimisation

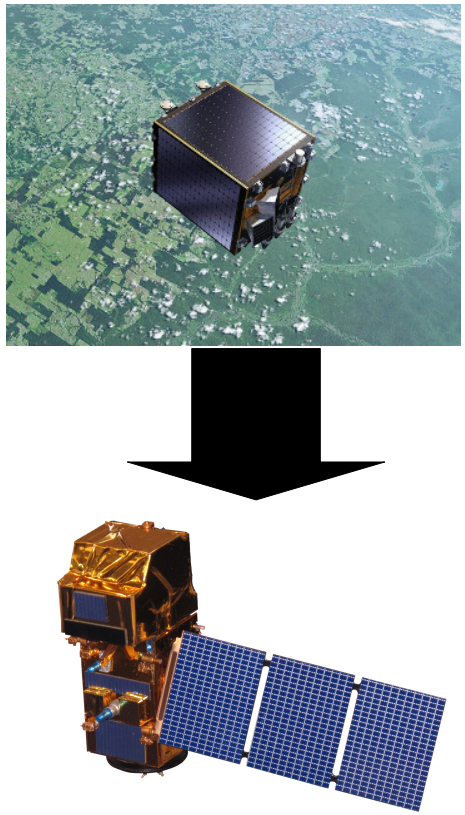
- We can detect changes, but how good can we do that?
- Change reference data by IIASA and WUR
- Optimised the parameters of BFAST and BFAST Monitor using global data
- Generally overestimates change (BFAST Monitor more so)
- Pairing with classifier output and expert rules needed to further reduce spurious change





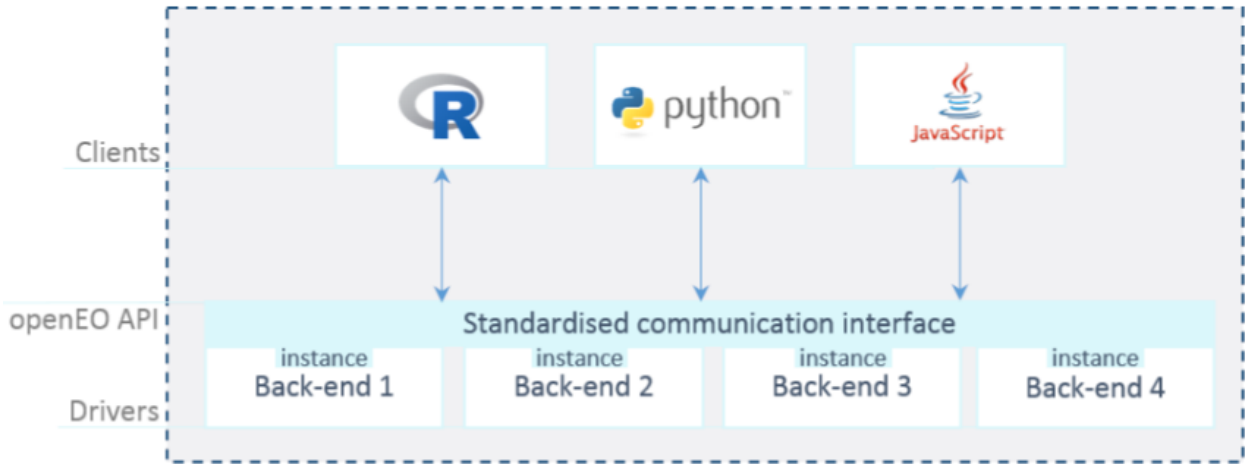
Future outlook: big data challenges

- Scaling down to 20 m
  - Sentinel-2 (20 m) instead of Proba-V (100 m), 25x
  - Landsat (30 m) instead of MODIS (250 m), 70x
  - Add Sentinel-1 20 m data for gap filling
- A cluster is nice, but even that is insufficient (and doesn't have the data)
  - Google Earth Engine?
  - Amazon?
  - DIASes?
  - BFAST on GPUs?



OpenEO

- Framework for handling large amounts of EO data
- Using a client (R/Python/JavaScript), can write a script that generates a language-agnostic process graph that is sent to a backend (GEE/GRASS/WCPS/JEODPP/GeoPySpark etc.)
- The backend accesses data locally (VITO also allows remote), runs the process graph and returns results: download only what you need
- Can run UDFs!
  e.g. BFAST



Other things I work on

- Geoscripting course: https://geoscripting-wur.github.io
- Lecturing (Master of Geo-information Science and Remote Sensing)
- SENSECO project and sun-induced fluorescence
    - Time series analysis
    - Point-based hyperspectral measurements from drones for photosynthesis efficiency and plant stress
- BFAST package maintenance

Thank you for your attention!

To explore the potential of nature to improve the quality of life

Copernicus Europe's eyes on Earth

Land Monitoring Service

WAGENINGEN UNIVERSITY & RESEARCH

vito

IIASA

Practicals

- Explore CGLS-LC100:

  https://lcviewer.vito.be

- Make your own land cover map:

  https://code.earthengine.google.com/?accept_repo=users/GreatEmerald/opengeohub2020

  Extra/inspiration: https://code.earthengine.google.com/?accept_repo=users/vitorsveg/scripts

- Detect breaks in time series:

  https://verbe039.github.io/BFASTforAEO/

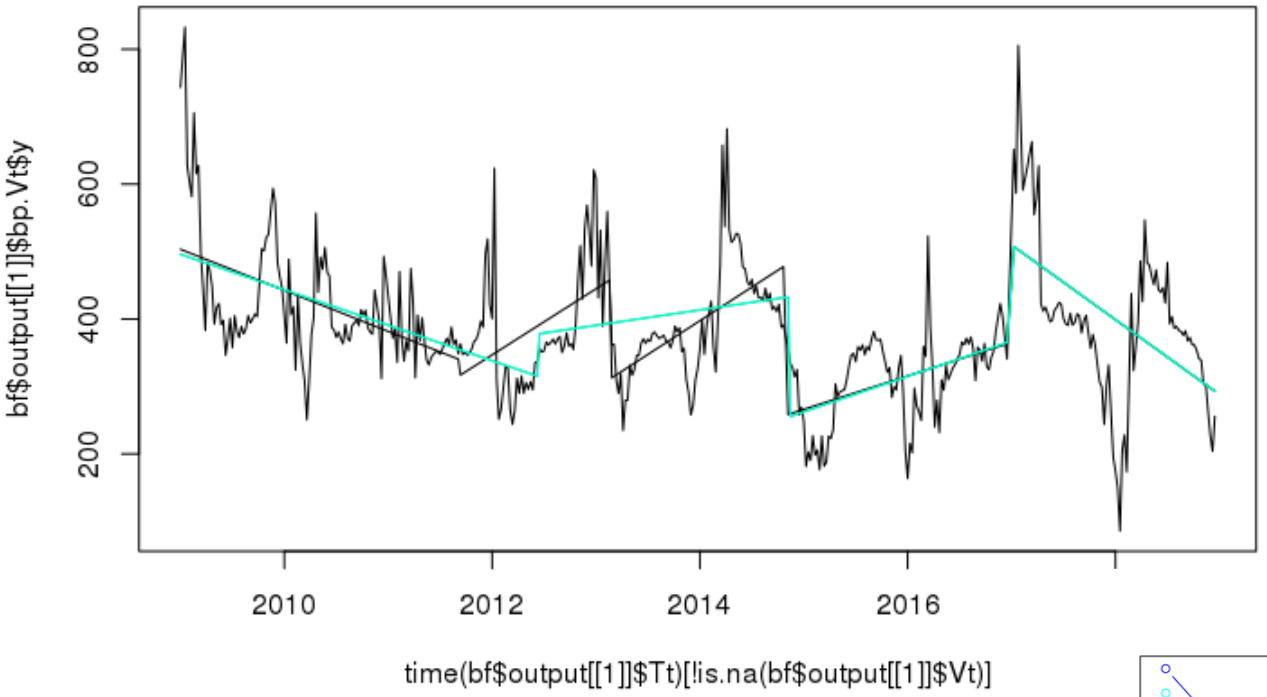
BFAST0N

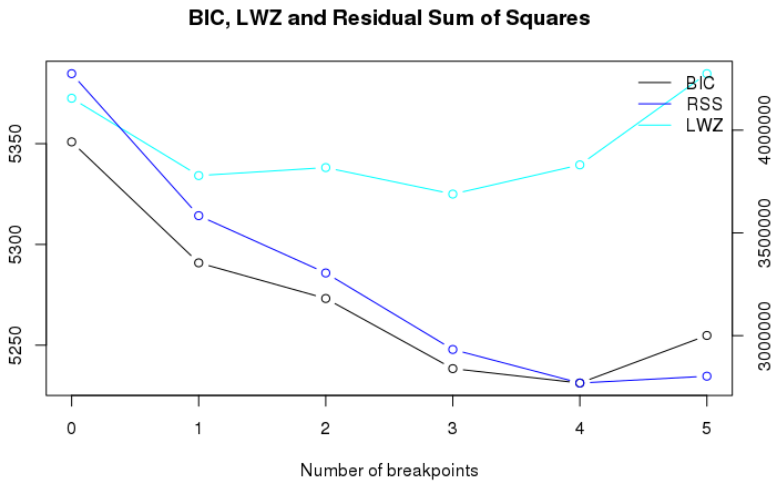
- In the development version of BFAST: install_github(“bfast2/bfast”)
- bfast::bfastpp(), +
- strucchange::breakpoints(formula, data, h)
  - data: any data.frame/matrix with numbers or `ts`
  - formula: e.g. response ~ trend + harmon
  - h: minimum segment size, either fraction of the time series length or integer defining the number of samples
- Output: `breakpointsfull` that indicates breakpoint timing and confidence interval, in sample numbers (mapping to `data`)

Principle of breakpoints()

- Piece-wise linear regression:
  - Given that we want one break, what's the optimal location to put it so that the RSS of two segments is minimised?
  - What if we want two breaks?
  - Etc. etc. to get a triangular matrix of possible breaks and model RSS
- But how many breaks does the time series have?
  - An Information Criterion: if we increase degrees of freedom by adding breaks, data will fit better, so penalise for each degree of freedom added
  - AIC (k=2) is too weak, BIC (k=log(n)) is also often too weak
  - LWZ ($k=0.299 \times \log(n)^{2.1}$) seems to do better

Breakpoints using LWZ vs BIC

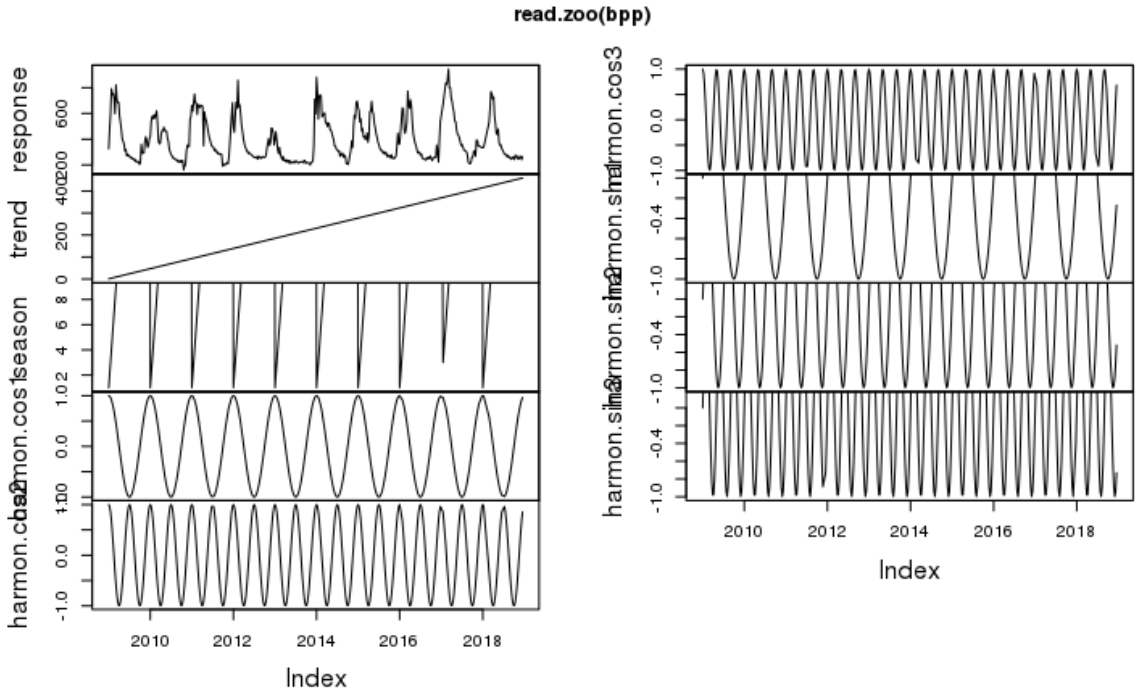




WAGENINGEN
UNIVERSITY & RESEARCH

bfastpp()

- How to get data with response ~ trend + harmon?
- bfastpp(ts, order): preprocessing of time series
  - ts must be a `ts` with frequency > 1
  - order is the harmonic order
- Output is a data.frame with:



New in BFAST0N

- Ability to use LWZ for selecting breaks
- Extra information when printing the results:
  - LWZ statistics
  - R^2
  - Break magnitude, using difference between segment models and the difference in last/first predicted value
- Parameter for customisable seasonal dummy number