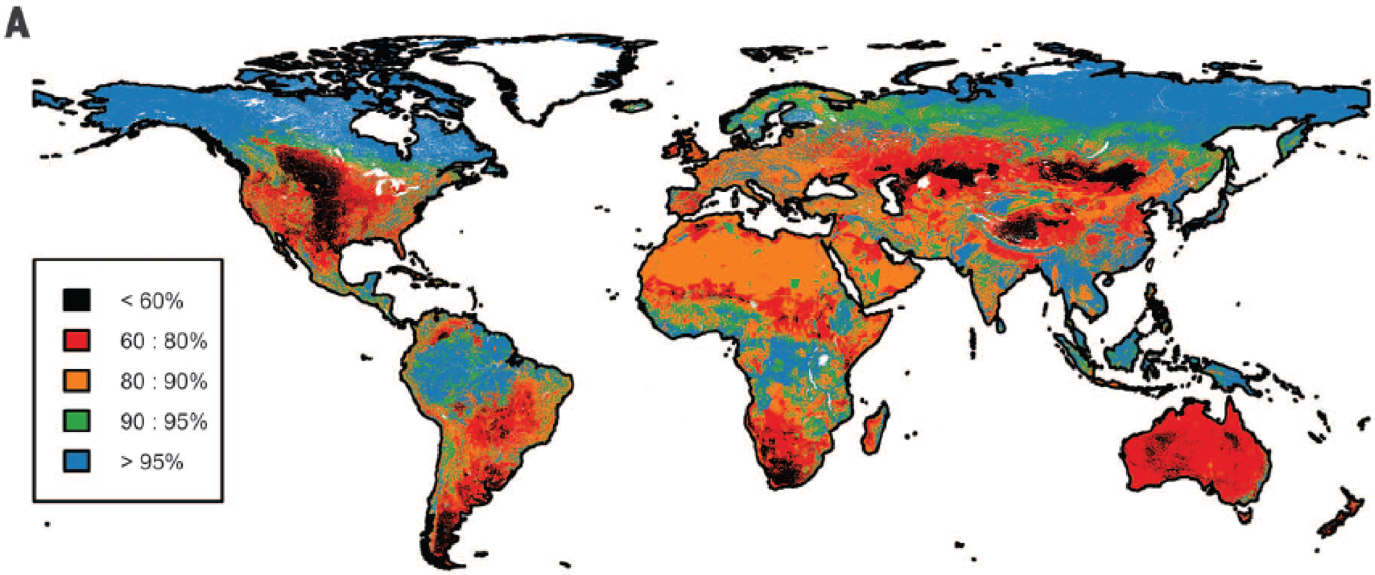
THE TIMES THEY ARE A CHANGIN'
MONITORING LAND DYNAMICS FROM SPACE

Martin Jung
m.jung@sussex.ac.uk

MY BACKGROUND

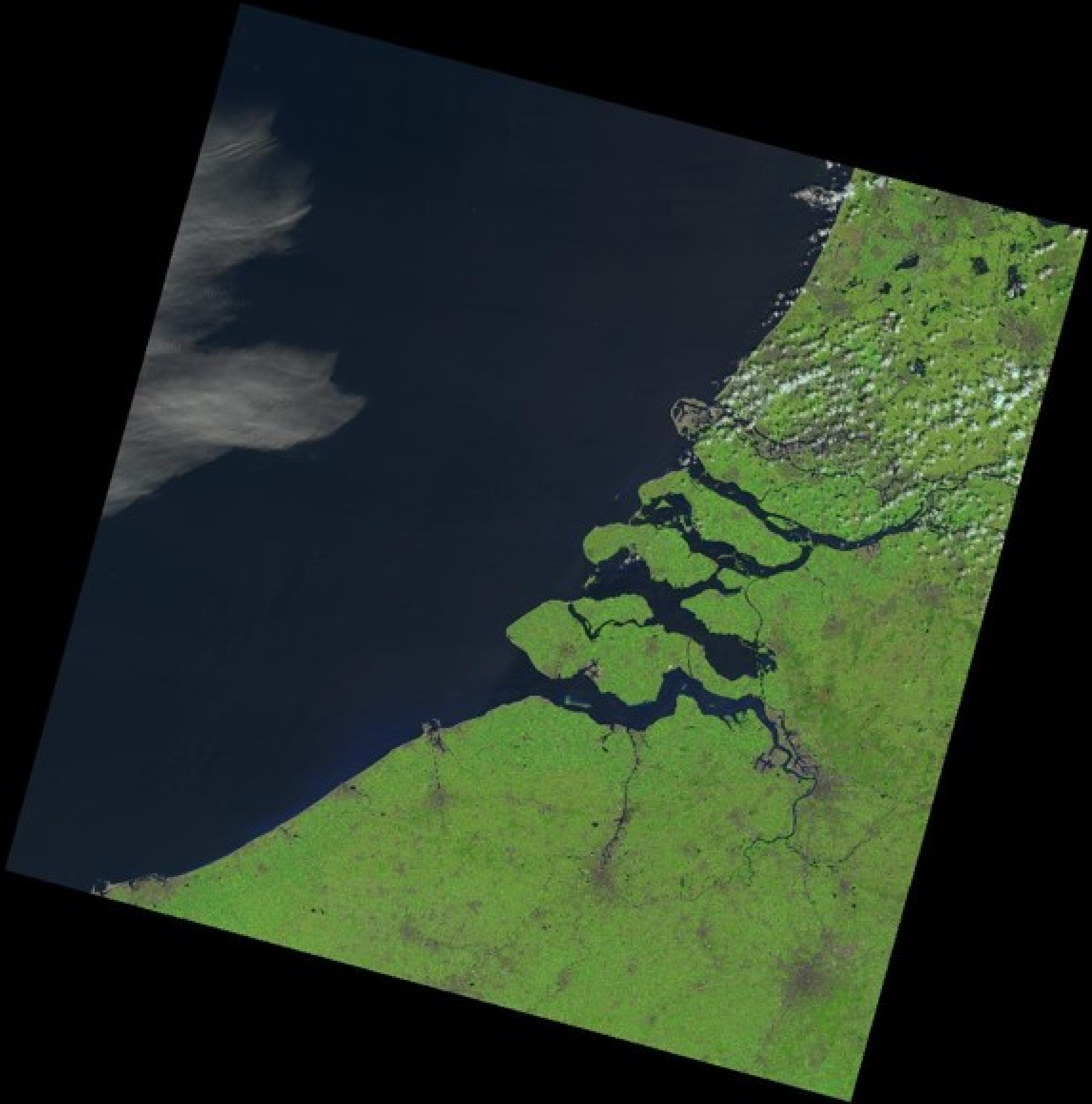
GLOBAL BIODIVERSITY MODELLING



Newbold et al. (2016)

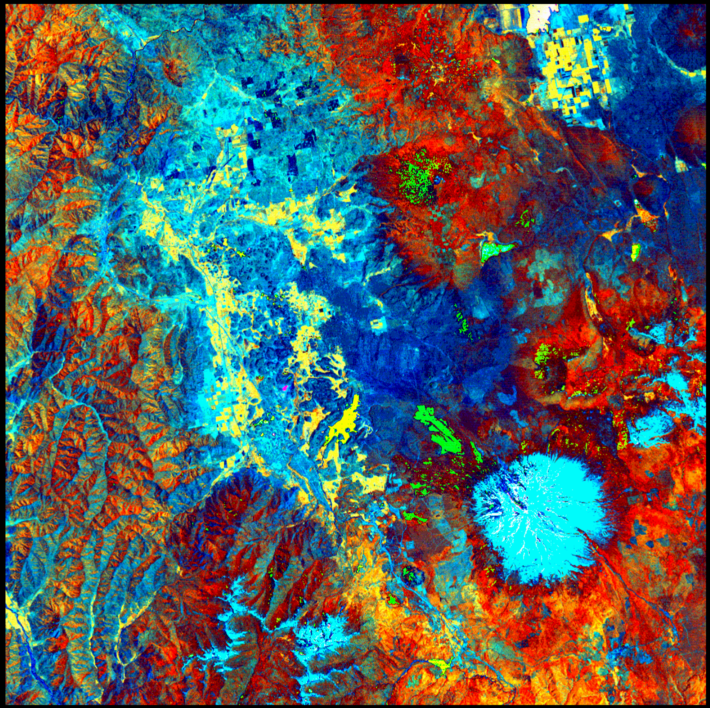
DOI: 10.1126/science.aaf2201

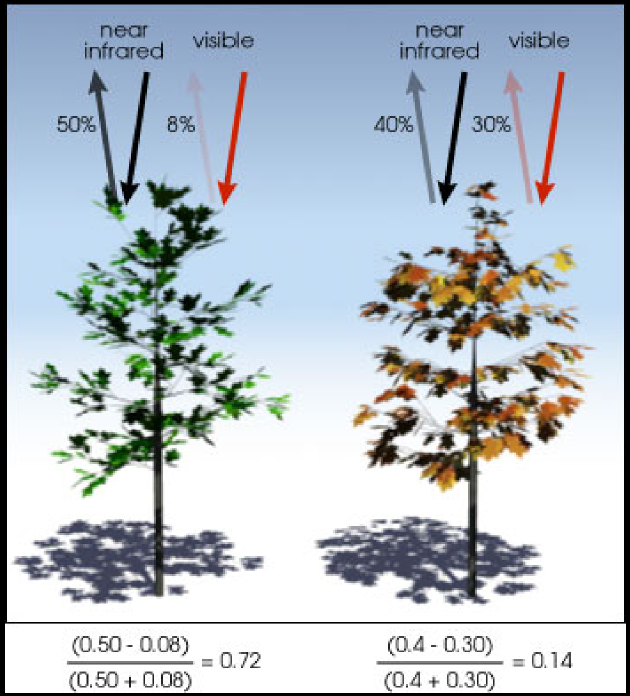
SATELLITE DATA



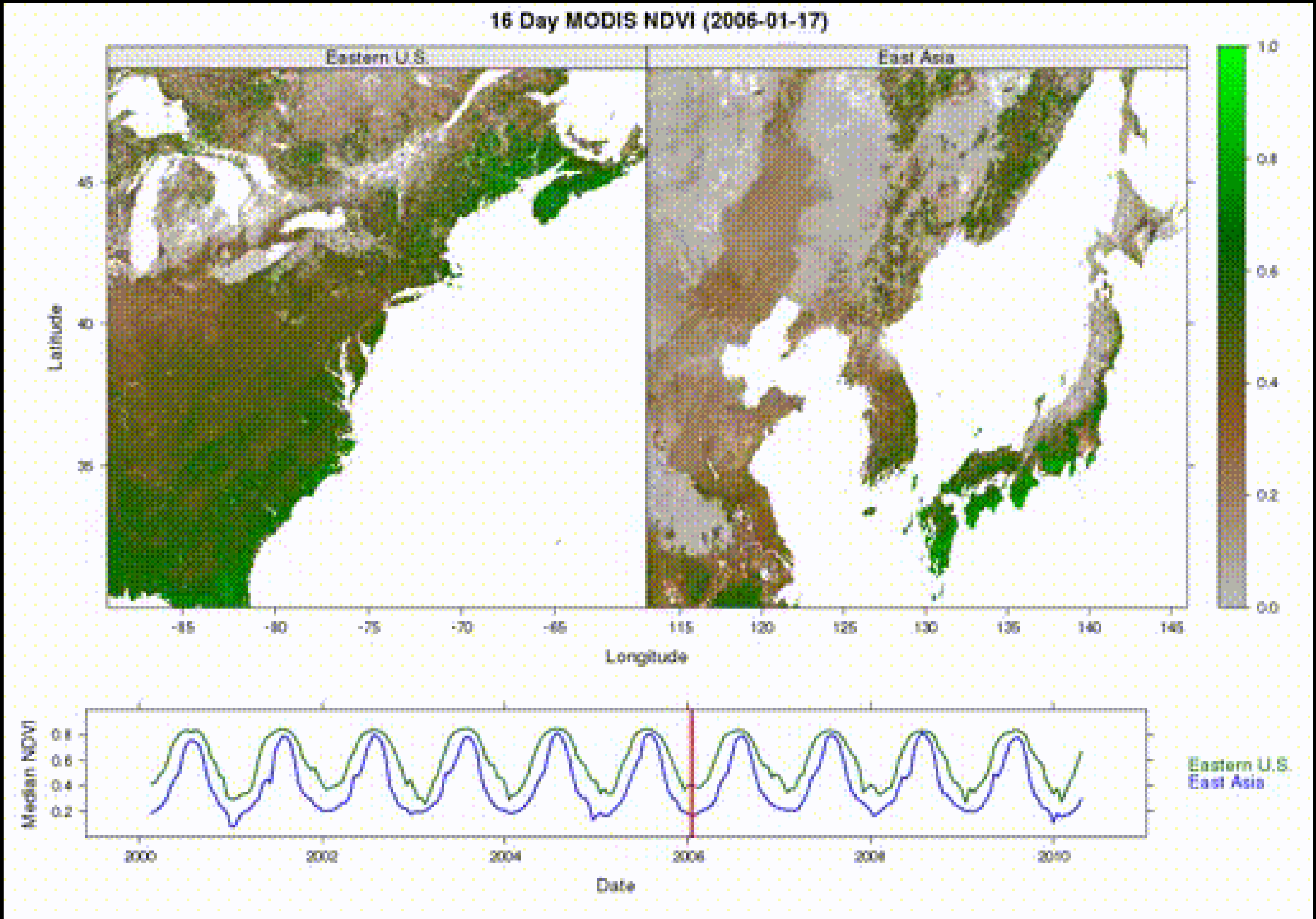

1998 190
10 km

MULTISPECTRAL IMAGES





VEGETATION PHENOLOGY



WHAT IS CHANGE ?

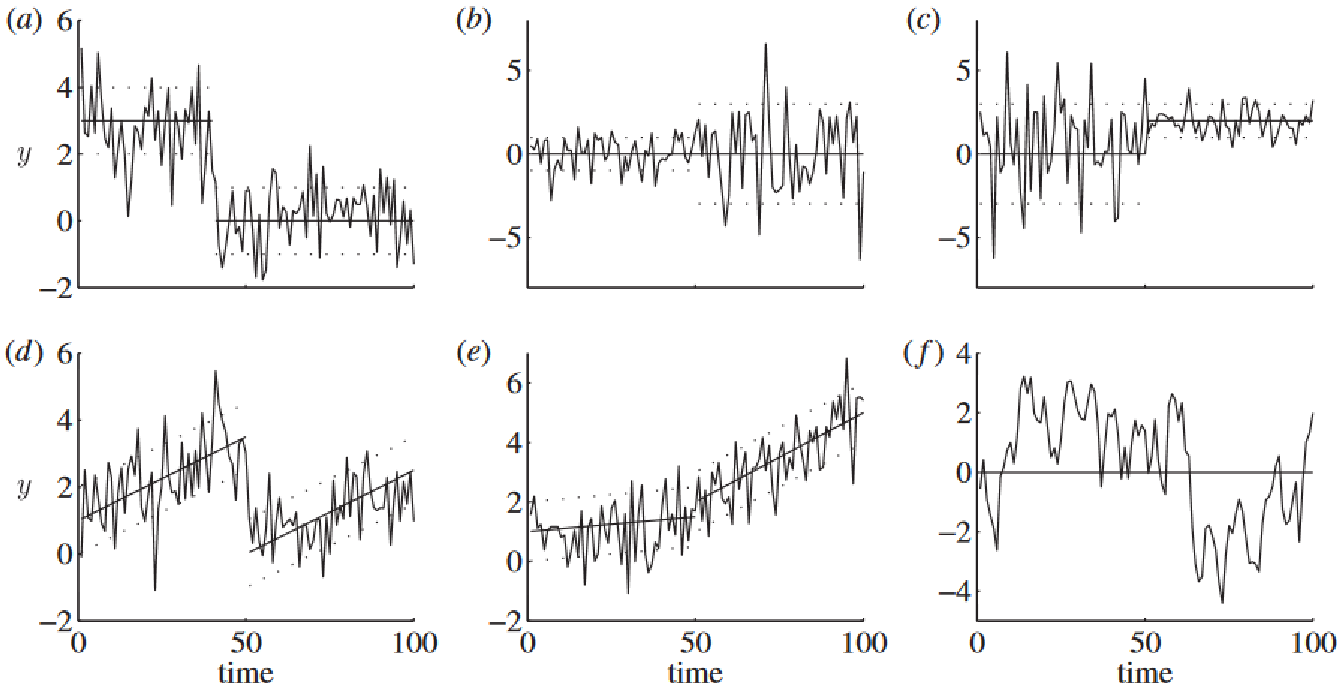


Figure 1. Examples of time series with a change point in (*a*) the mean, (*b*) the variance, (*c*) both the mean and the variance, (*d*) the intercept of a linear regression model and (*e*) both the intercept and the trend of a linear regression model, and (*f*) no change point, but a strong positive autocorrelation.

Beaulieu *et al* (2012) doi: [10.1098/rsta.2011.0383](https://doi.org/10.1098/rsta.2011.0383)

CHANGES IN TREND

BFAST

Breaks For Additive Seasonal and Trend

Verbesselt et al. (2010) doi: 10.1016/j.rse.2010.08.003

Decomposes and assess change in trend, seasonal and remainder term of time series

R: *bfast*, *strucchange*

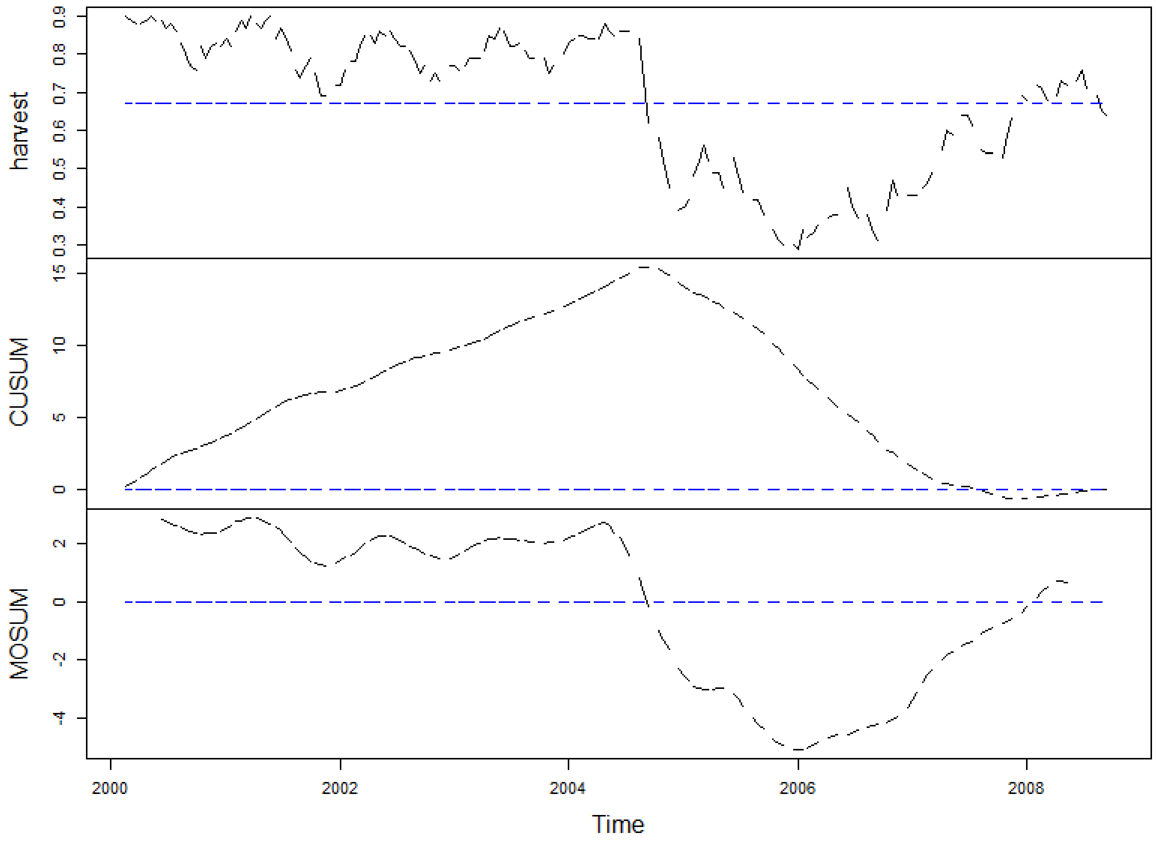
$$Y_t = T_t + S_t + e_t$$
$$(t = 1, \ldots, n)$$

$$T_t = \alpha_i + \beta_i t$$
$$\tau^*_{j-1} < \tau \leq \tau^*_j \quad (j = 1, \ldots, m)$$

$$S_t = \sum_{k=1}^{K} a_{j,k}\ sin\left(\frac{2\pi kt}{f} + \delta_{j,k}\right)$$

$$\begin{aligned} K &= \text{Harmonic terms} \\ a_{j,k} &= \text{Amplitude} \\ \delta_{j,k} &= \text{Phase} \\ f &= \text{Frequency} \end{aligned}$$

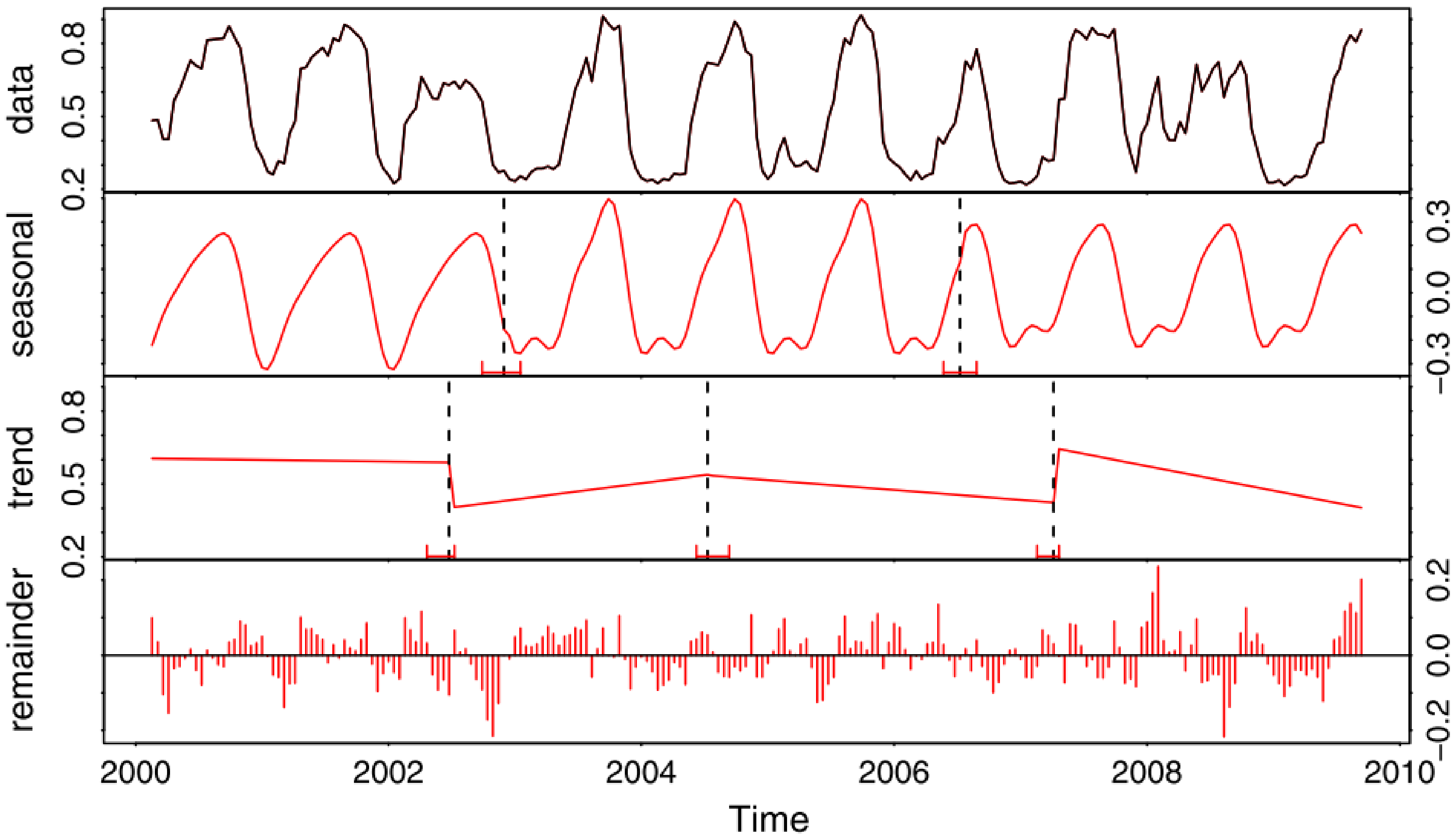
$$\tau_{j-1}^{\#} < \tau \leq \tau_{j}^{\#} \quad (j = 1, \ldots, p)$$



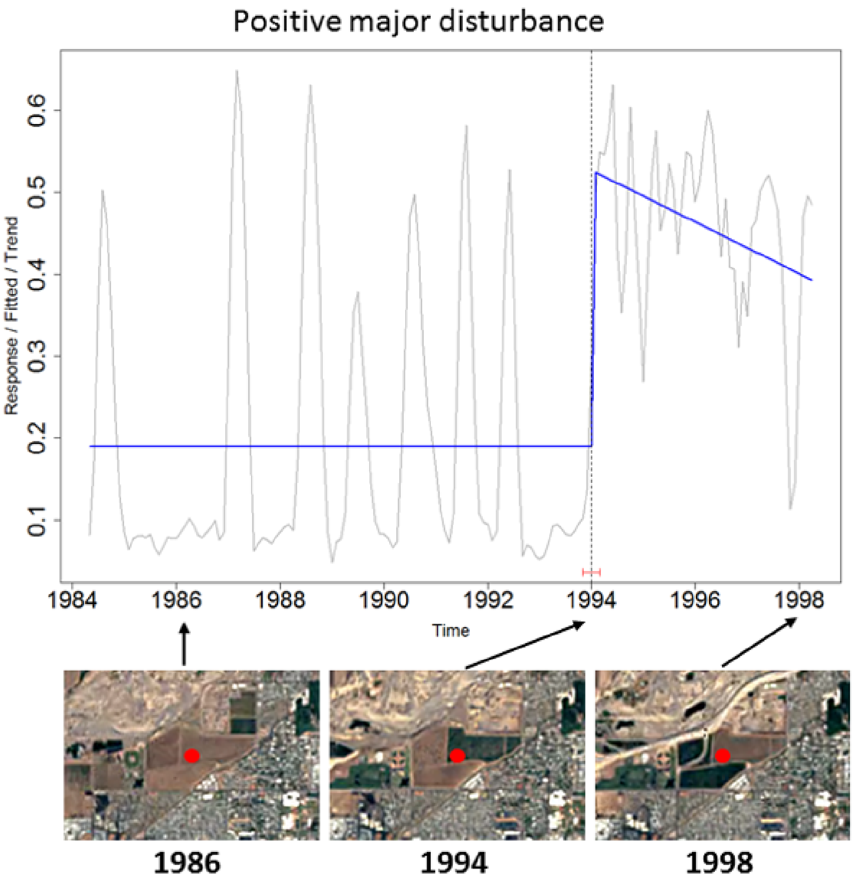
$$MO_t = \frac{1}{\delta\sqrt{n}} \sum_{s=t-h+1}^{t} (y_s - \hat{y}_s)$$

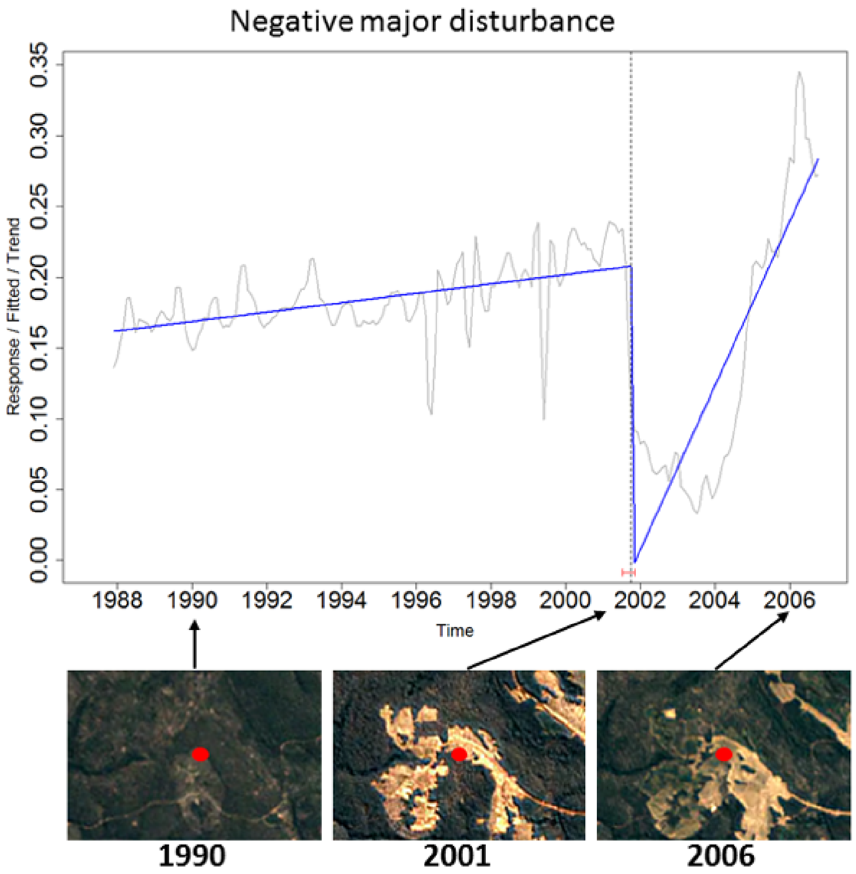
1. If test indicates presence of breakpoints: Estimate number and position in
$$Y_t - S_t$$
and
$$Y_t - T_t$$
2. Reiterate until est. parameters remain unchanged.

J. Verbesselt et al. / Remote Sensing of Environment 114 (2010) 2970–2980



DOES IT WORK ? SORT, OF...





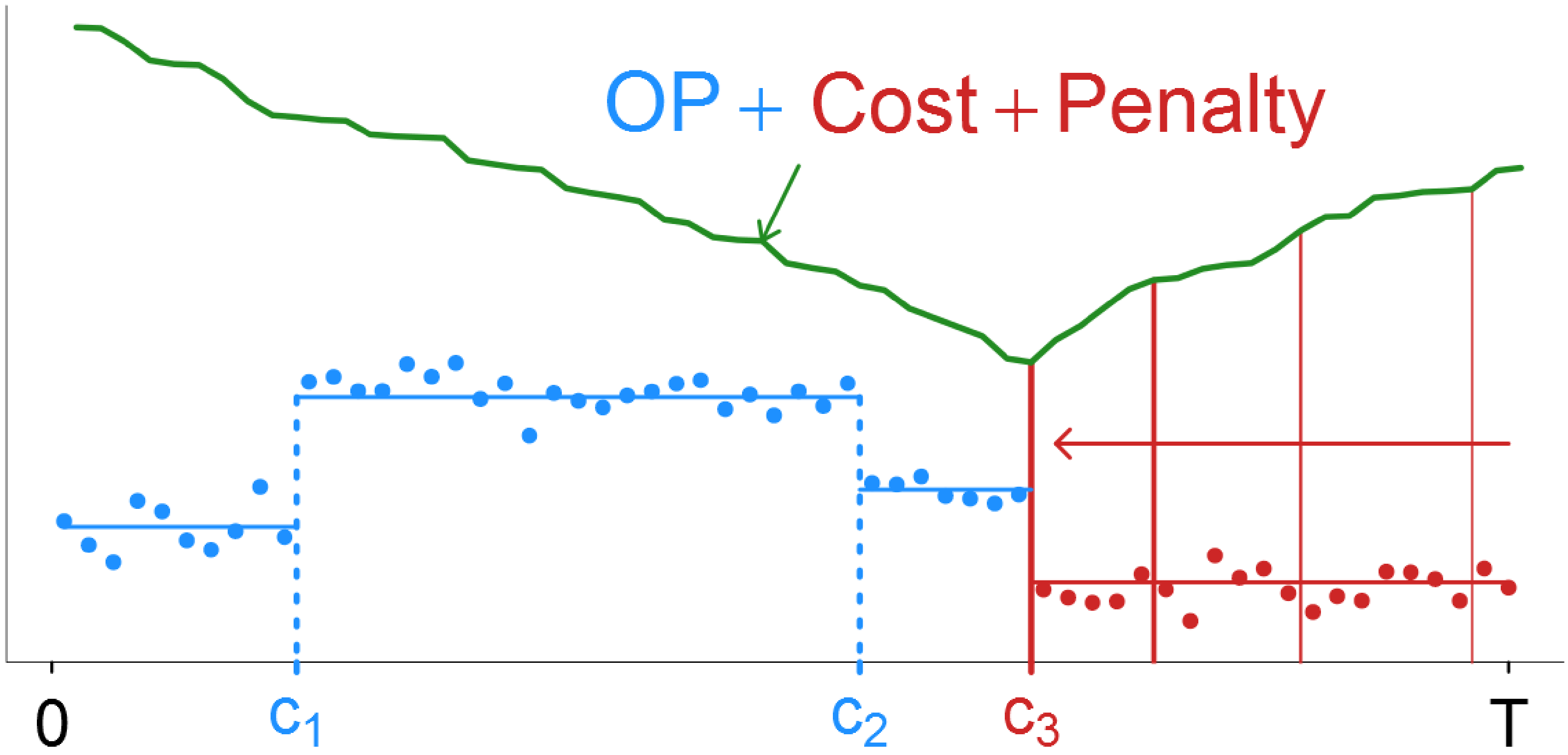
CHANGES IN MEAN + VARIANCE

Optimal detection of changepoints with a linear computational cost

Killick et al. (2012) doi: 10.1080/01621459.2012.737745

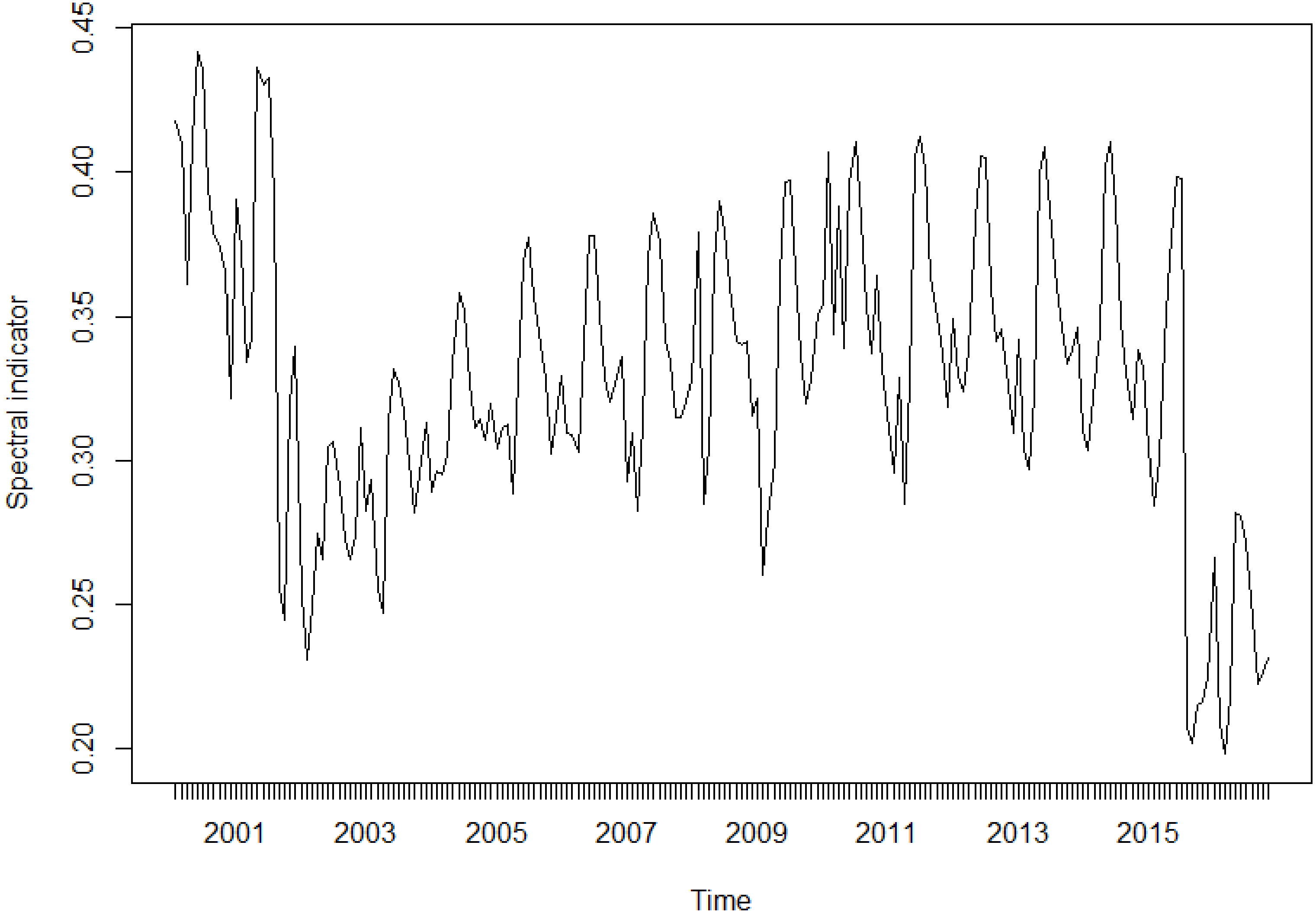
$$\sum_{i=1}^{m+1}\left[C\left(y_{(\tau_{i-1}+1):\tau_i}\right)\right] + \beta f(m)$$

$$\begin{aligned} C &= \text{Cost function for segment} \\ \beta f(m) &= \text{Penalty term} \end{aligned}$$

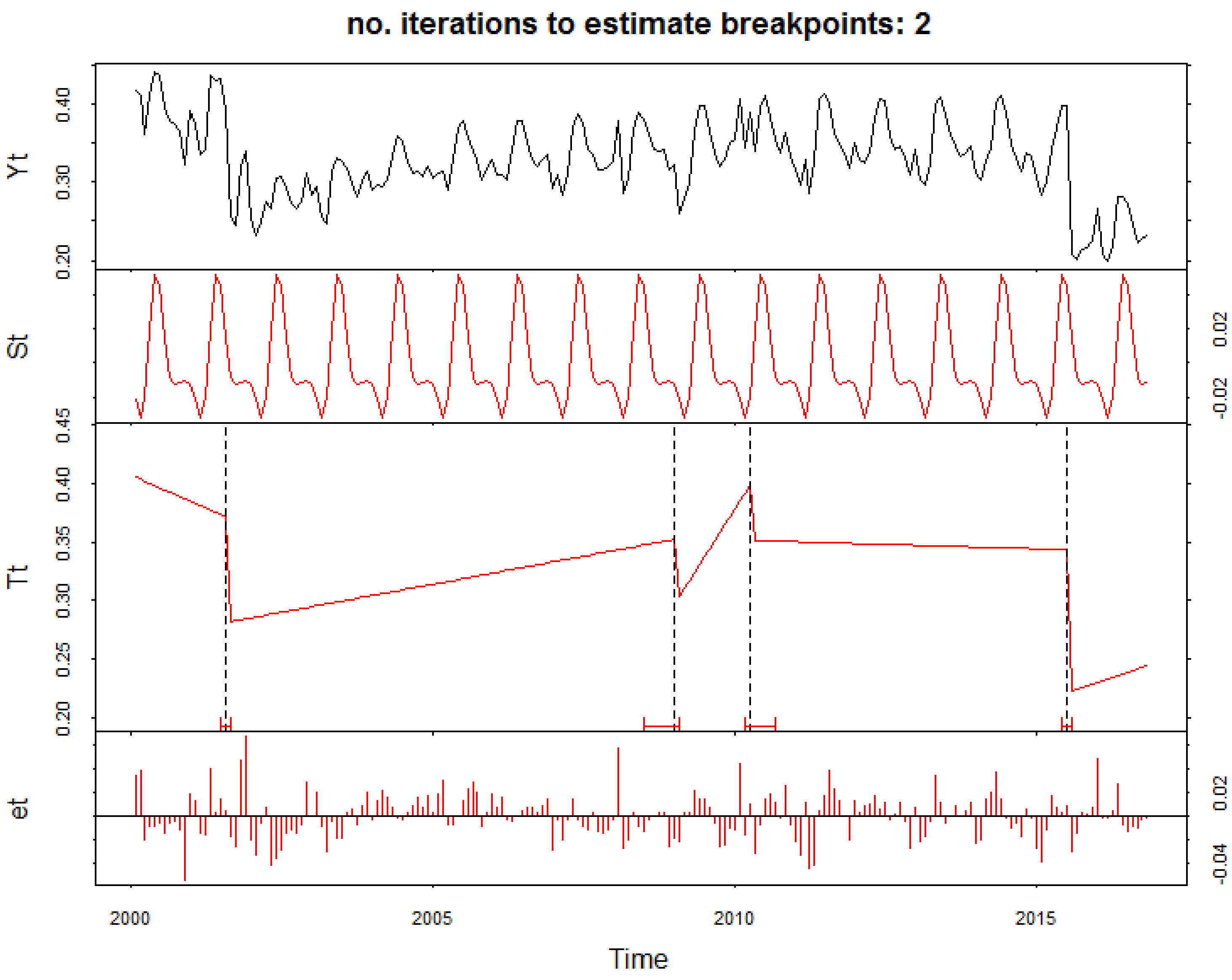


(Source: Michael Messer - Uni Frankfurt)

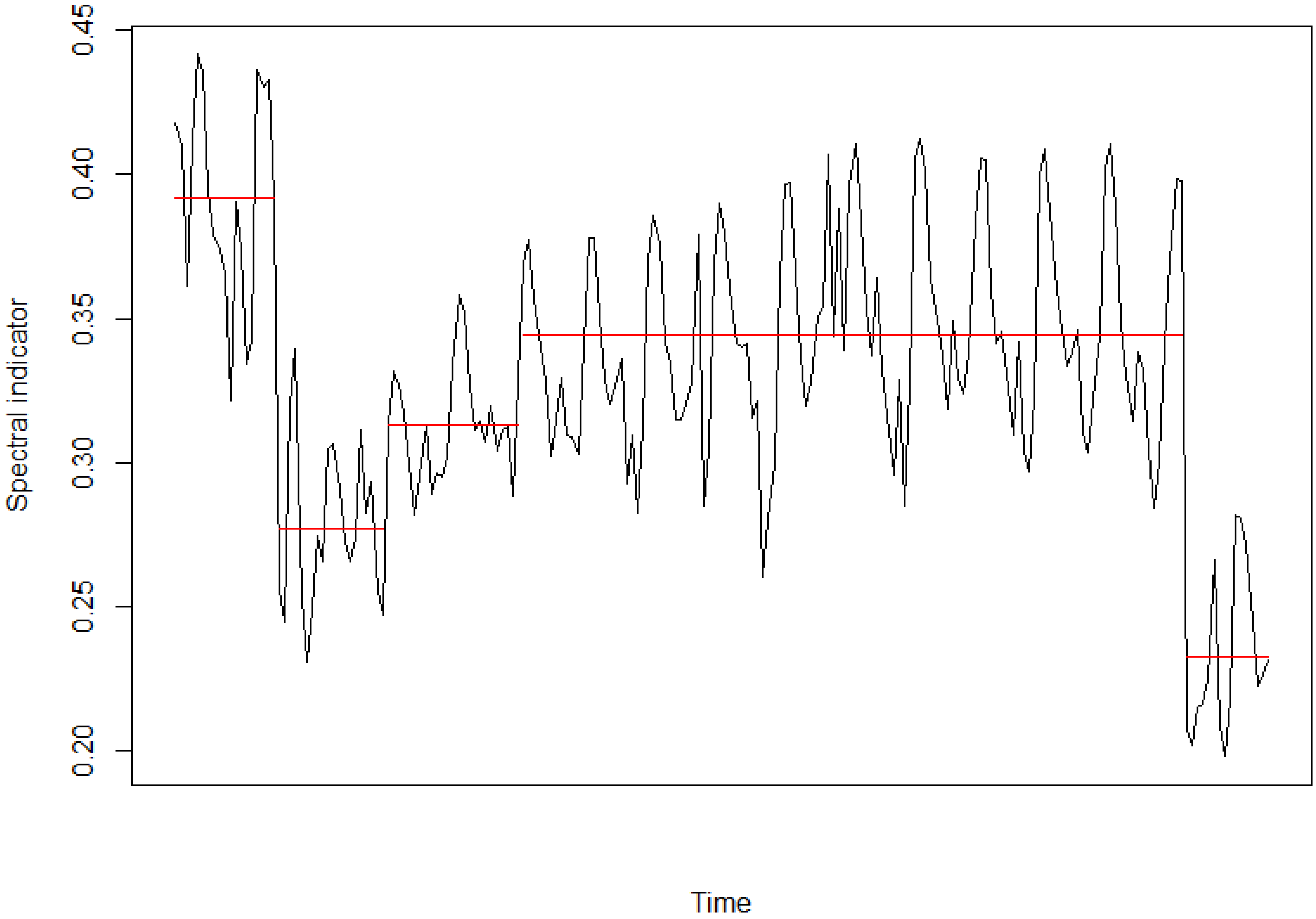
DOES IT WORK ?



Spectral indicator
0.20
0.25
0.30
0.35
0.40
0.45
2001
2003
2005
2007
2009
2011
2013
2015
Time



no. iterations to estimate breakpoints: 2
Yt
St
Tt
et
0.20
0.30
0.40
0.02
-0.02
0.20
0.25
0.30
0.35
0.40
0.45
0.02
-0.04
2000
2005
2010
2015
Time

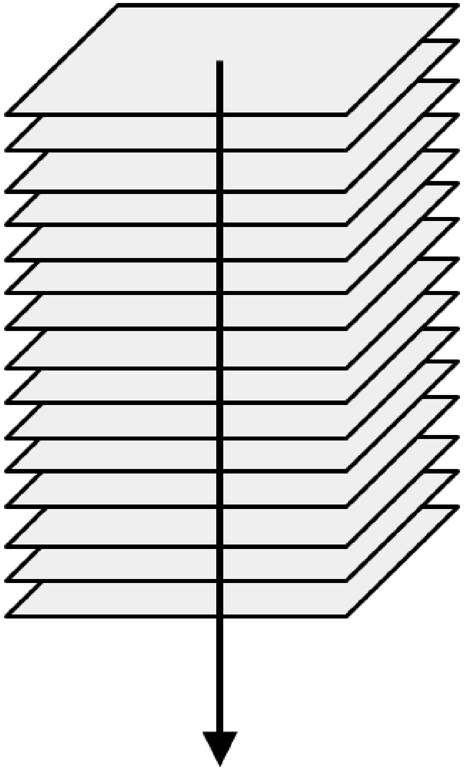


Spectral indicator
0.45
0.40
0.35
0.30
0.25
0.20
Time

AN IDEAL BREAKPOINT DETECTION?

- Detects both changes in variance and abrupt breaks
- Can test for none, single and multiple breaks
- Works with missing data!
- robust to noise and accounts for seasonal fluctuations
- multivariate measurements of ordered observations ?
- Execution time under 5s ideally

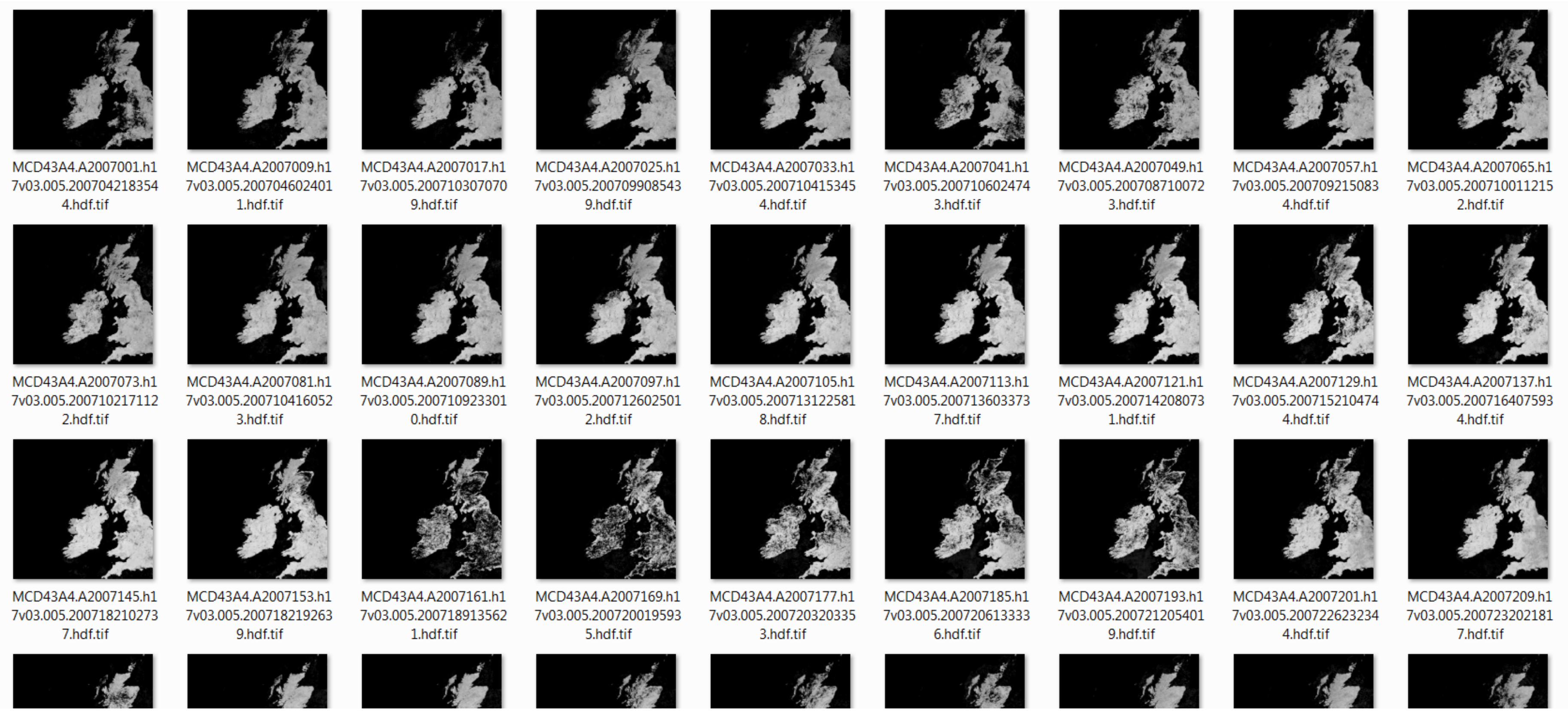
CHALLENGES OF WORKING WITH SPATIAL DATA



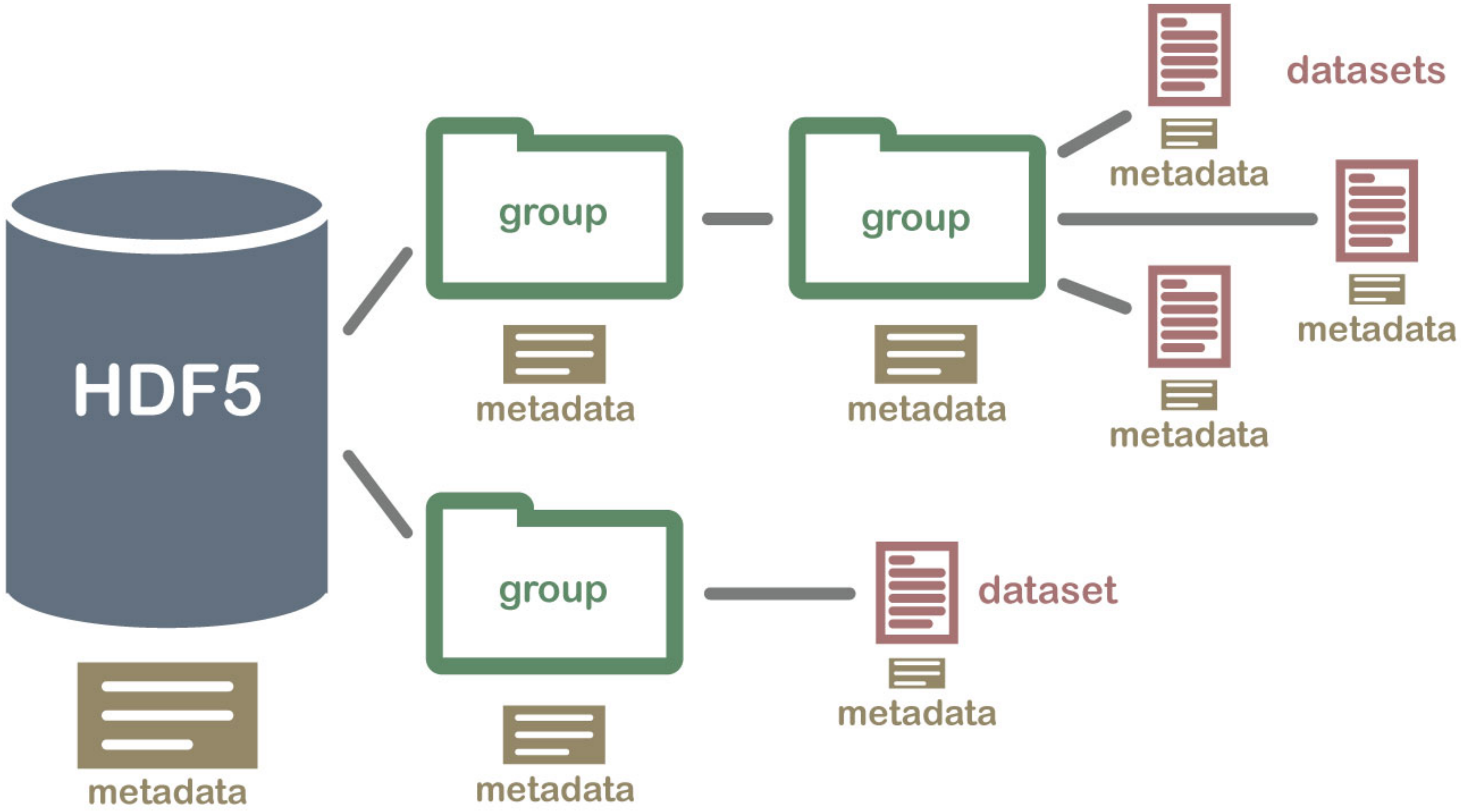
Multi-dimensional arrays x * y * z

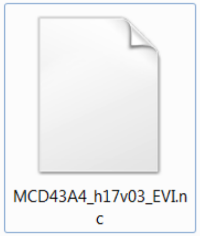
Big data (46 * 32y = ~1472GB)

MANANGEMENT OF LARGE NUMBERS OF FILES



INSTEAD USE A HIERARCHICAL FILE FORMAT



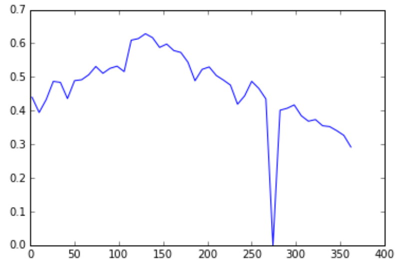


Compression ratio: **64.2 %**

```
root group (NETCDF4 data model, file format HDF5):
    title: MCD43A4
    description: contains EVI MCD43A4 for the UK
    institution: University of Sussex, United Kingdom
    source: based on MODIS Terra & Aqua - BRDF Product
    year: 2007
    projection: WGS84
    dimensions(sizes): lat(2400), lon(2400), time(46)
    variables(dimensions): float32 time(time), float32 la
```

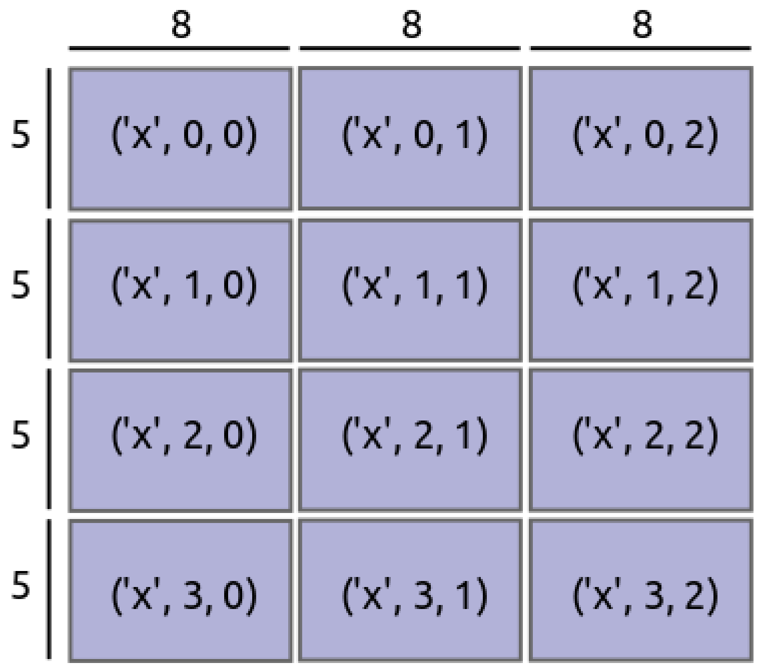
```
import xray
import matplotlib.pyplot as plt
f = xray.open_dataset('MCD43A4_h17v03_EVI.nc')
fm = f.sel(lon=[50.86],lat= [-0.088029], method='nearest')
.groupby('time.dayofyear').mean()

# Plot
plt.plot(fm.dayofyear,fm.metric)
```



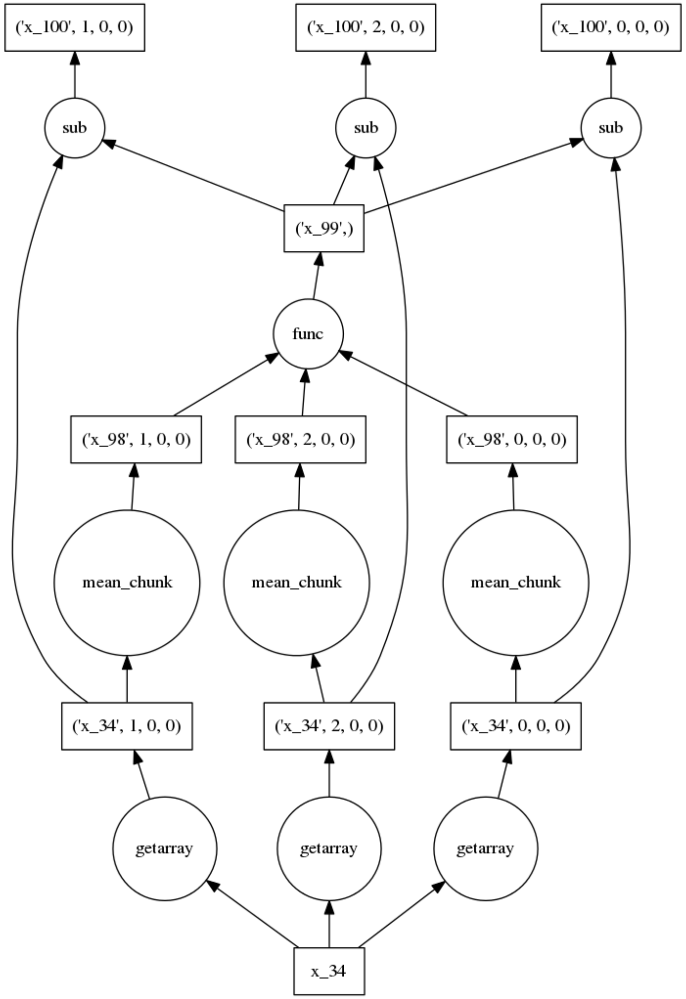
WHAT IF MY DATA IS TO LARGE TO LOAD INTO MEMORY AT ONCE ?

Dask divides arrays into many small pieces, called chunks, and provides out-of-core computation .



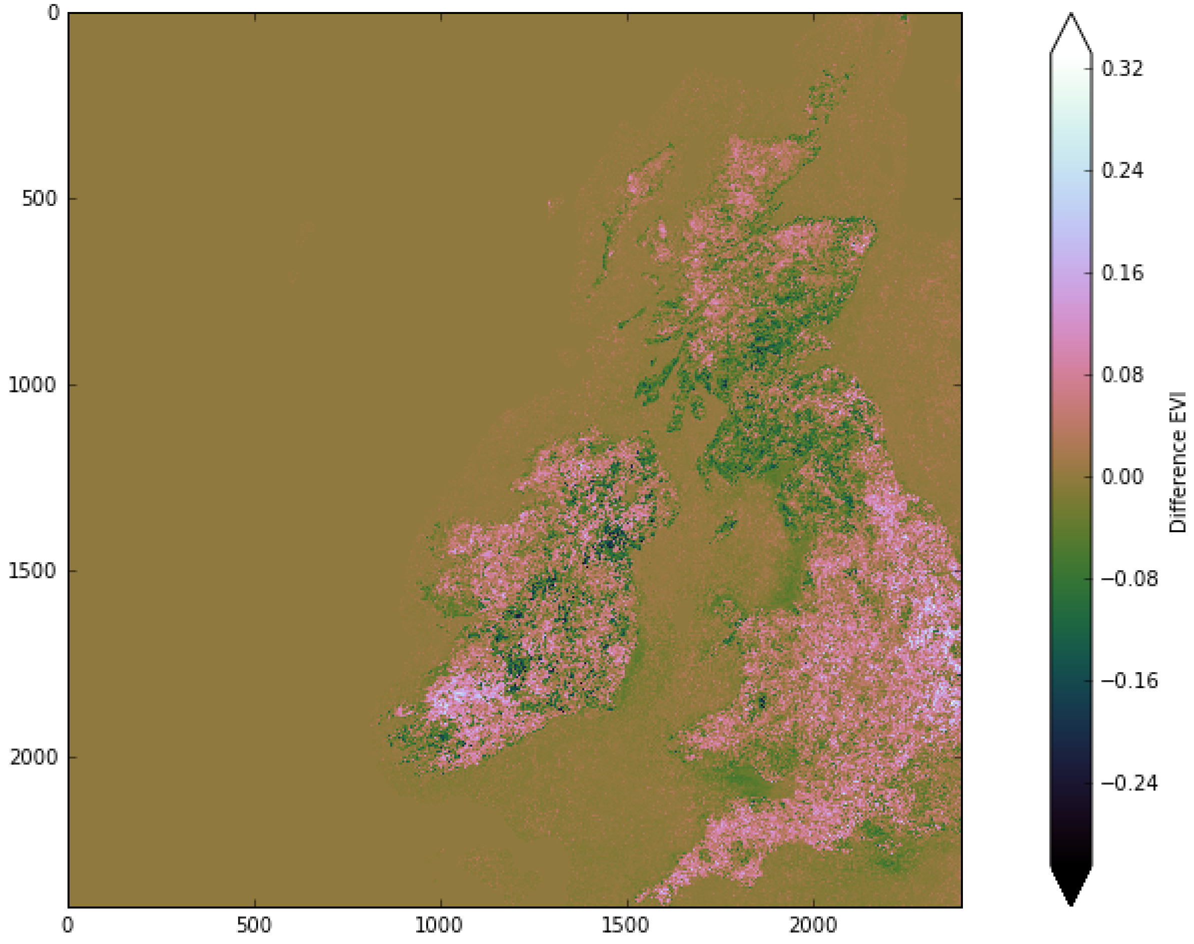
```
f = xray.open_dataset('MCD43A4_h17v03_EVI.nc',chunks=100)
f_1 = f.groupby('time.month').mean('time')
f_2 = f.mean('time') # Overall mean
full = (f_1.sel(month=3) - f_2).metric

# Now it gets computed
%time result = full.load()
```



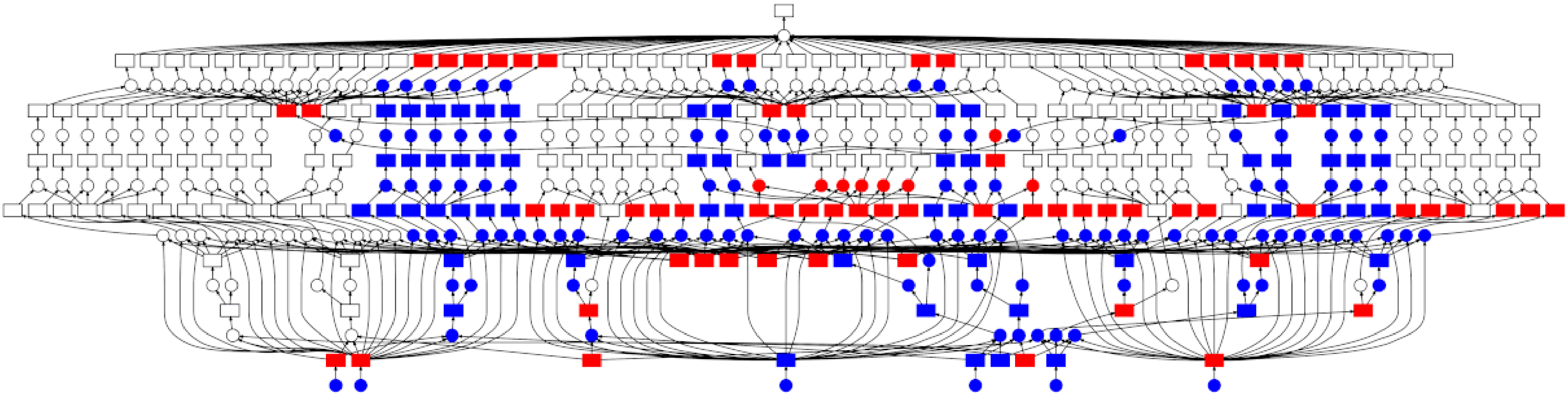
('x_100', 1, 0, 0)
('x_100', 2, 0, 0)
('x_100', 0, 0, 0)
sub
sub
sub
('x_99',)
func
('x_98', 1, 0, 0)
('x_98', 2, 0, 0)
('x_98', 0, 0, 0)
mean_chunk
mean_chunk
mean_chunk
('x_34', 1, 0, 0)
('x_34', 2, 0, 0)
('x_34', 0, 0, 0)
getarray
getarray
getarray
x_34

Total: Wall time: 17.7 s



EXTENDABLE FOR MULTIPLE CLIENT WORKERS

```
from distributed import Client
client = Client() # distributes work process over workers
```



INTERESTING? HELPFUL?