Supplement of Biogeosciences, 17, 945–962, 2020
https://doi.org/10.5194/bg-17-945-2020-supplement

Supplement of

Towards a global understanding of vegetation–climate dynamics at multiple timescales

Nora Linscheid et al.

Correspondence to: Nora Linscheid (nlinsch@bgc-jena.mpg.de) and Lina M. Estupinan-Suarez (lestup@bgc-jena.mpg.de)

Supplementary Figures

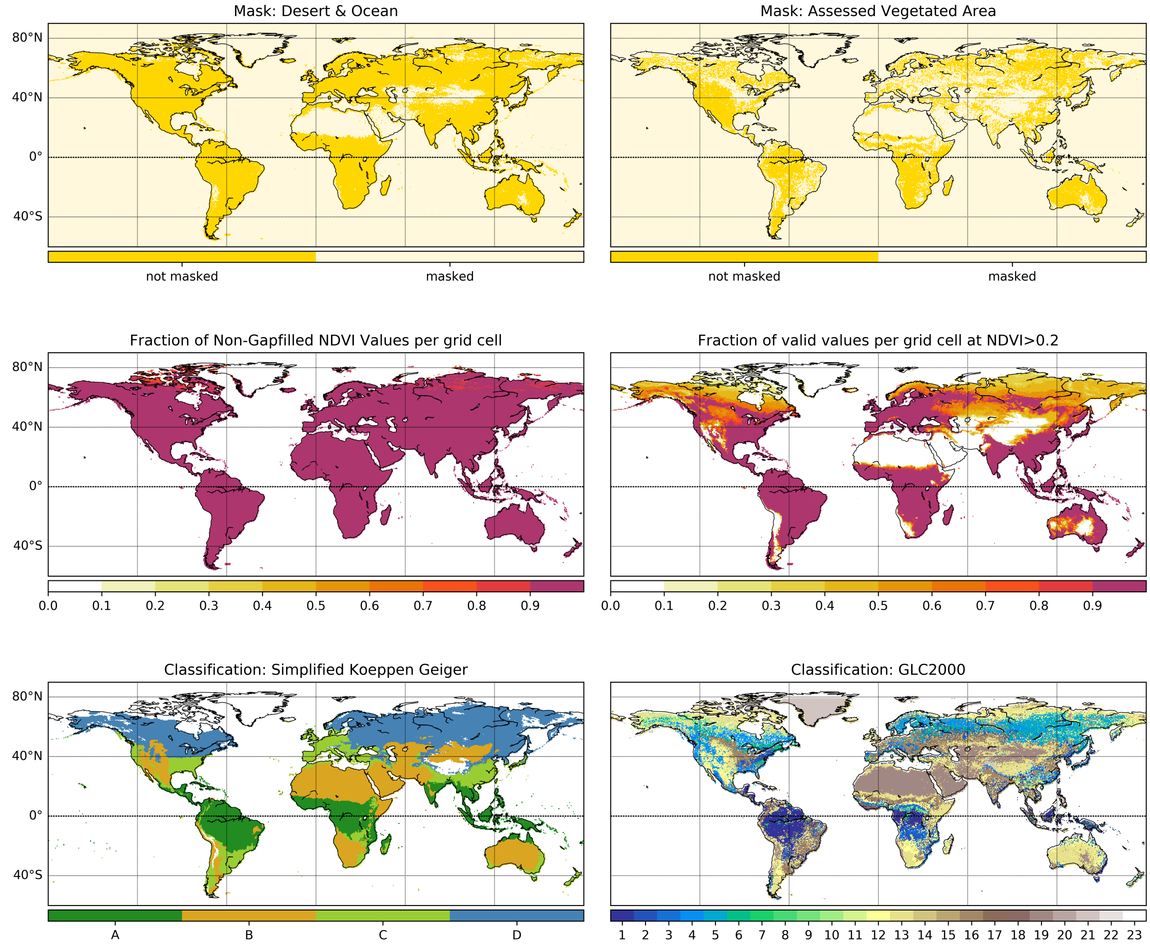


Figure S1: **Masks and Classification schemes used in the analyses.** **a.** Mask for deserts and oceans, **b.** Mask for natural vegetated area based on GLC2000, **c.** Fraction of non–gapfilled NDVI values per grid cell based on GIMMS NDVI, **d.** Fraction of valid values per grid cell after filtering for NDVI > 0.2, **e.** Simplified Köppen–Geiger Classification, A: equatorial, B: arid, C: warm temperate, D: snow, **f.** Classification of land cover classes after Global Land Cover 2000 (GLC2000). Numbers from 1–23 represent: 1 – Tree Cover broadleaved evergreen, 2 – Tree Cover broadleaved deciduous closed, 3 – Tree Cover broadleaved deciduous open, 4 – Tree Cover needle leaved evergreen, 5 – Tree Cover needle leaved deciduous, 6 – Tree Cover mixed leaf type, 7 – Tree Cover regularly flooded fresh water, 8 – Tree Cover regularly flooded saline water, 9 – Mosaic: Tree Cover and other natural vegetation, 10 – Tree Cover burnt, 11 – Shrub Cover closed open evergreen, 12 – Shrub Cover closed open deciduous, 13 – Herbaceous Cover closed open, 14 – Sparse herbaceous or sparse shrub cover, 15 – Regularly flooded shrub and or herbaceous cover, 16 – Cultivated and managed areas, 17 – Mosaic: Cropland Tree Cover Other natural vegetation, 18 – Mosaic: Cropland Shrub and or grass cover, 19 – Bare Areas, 20 – Water Bodies, 21 – Snow and Ice, 22 – Artificial surfaces and associated areas, 23 – no data.

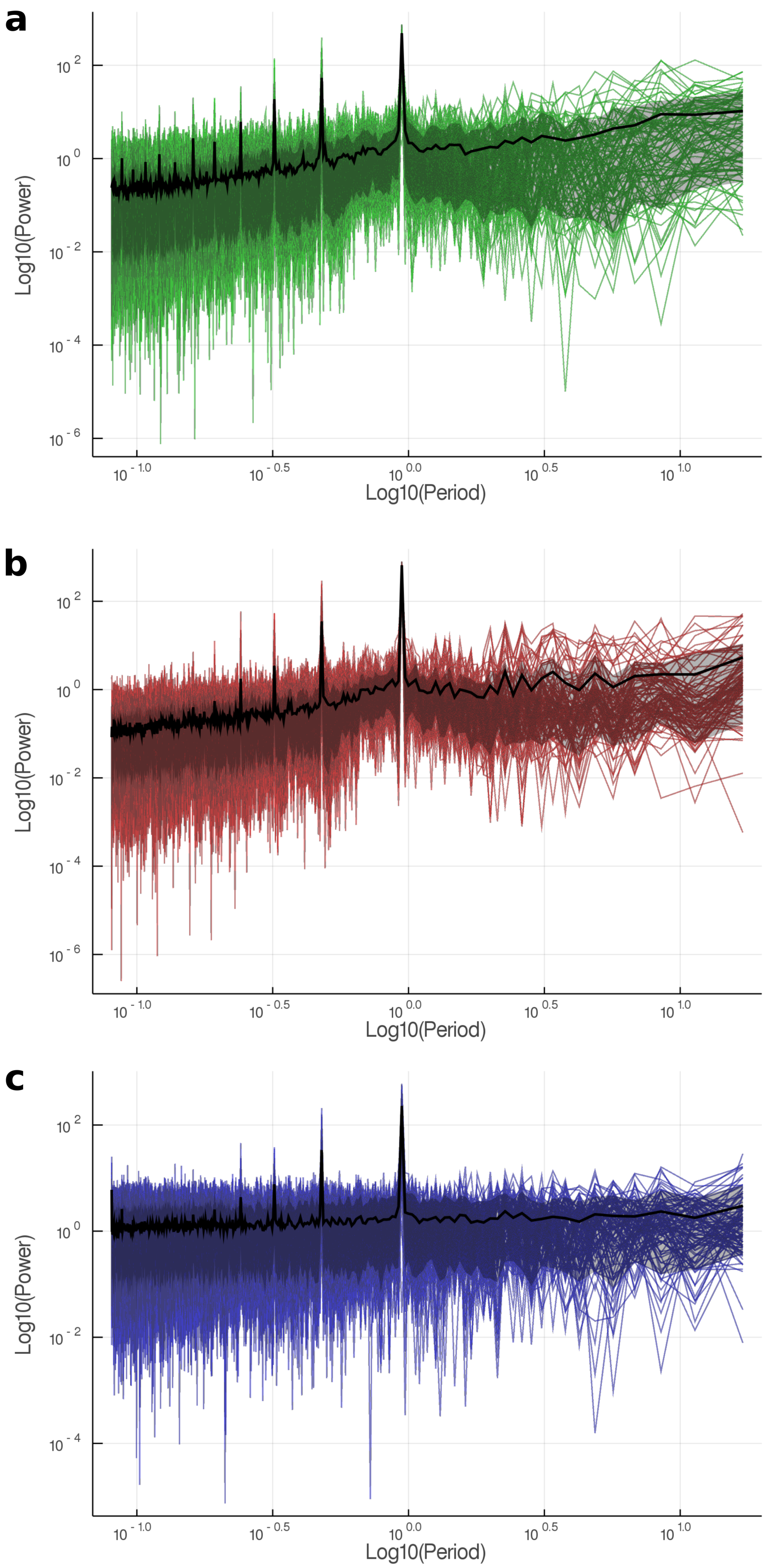


Figure S2: **Representative power spectra of Fourier decomposed NDVI (a), air temperature (b) and precipitation (c) time series.** Mean and 10th–90th percentile of power spectra are plotted as black line (mean) and band (percentiles), overlaid by 10000 sample spectra. Shortest signal periods (fastest frequencies) are plotted on the left side of the x–axis, longest periods on the right side of the x–axis. The annual period is located at 10^0.

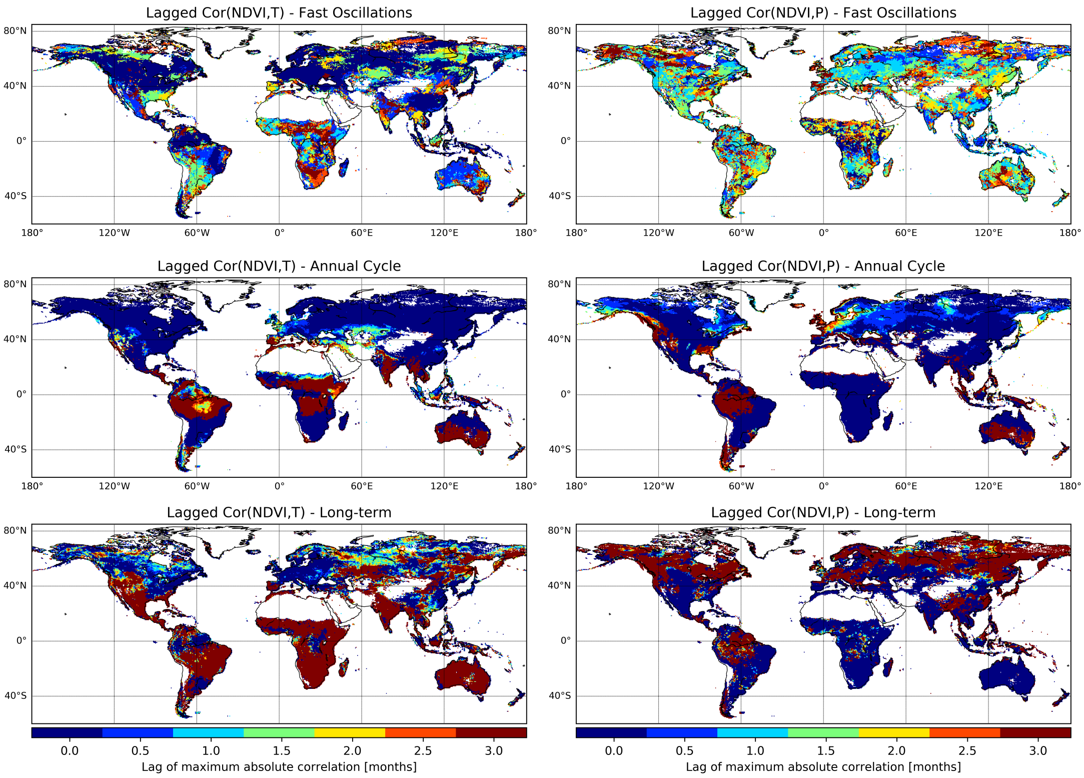


Figure S3: **Lag of maximum absolute correlation** between NDVI and air temperature (T, left panel) and NDVI and precipitation (P, right panel) at each grid cell. The time step used is 15 days, which is equivalent to a 0.5–month lag in the color key.

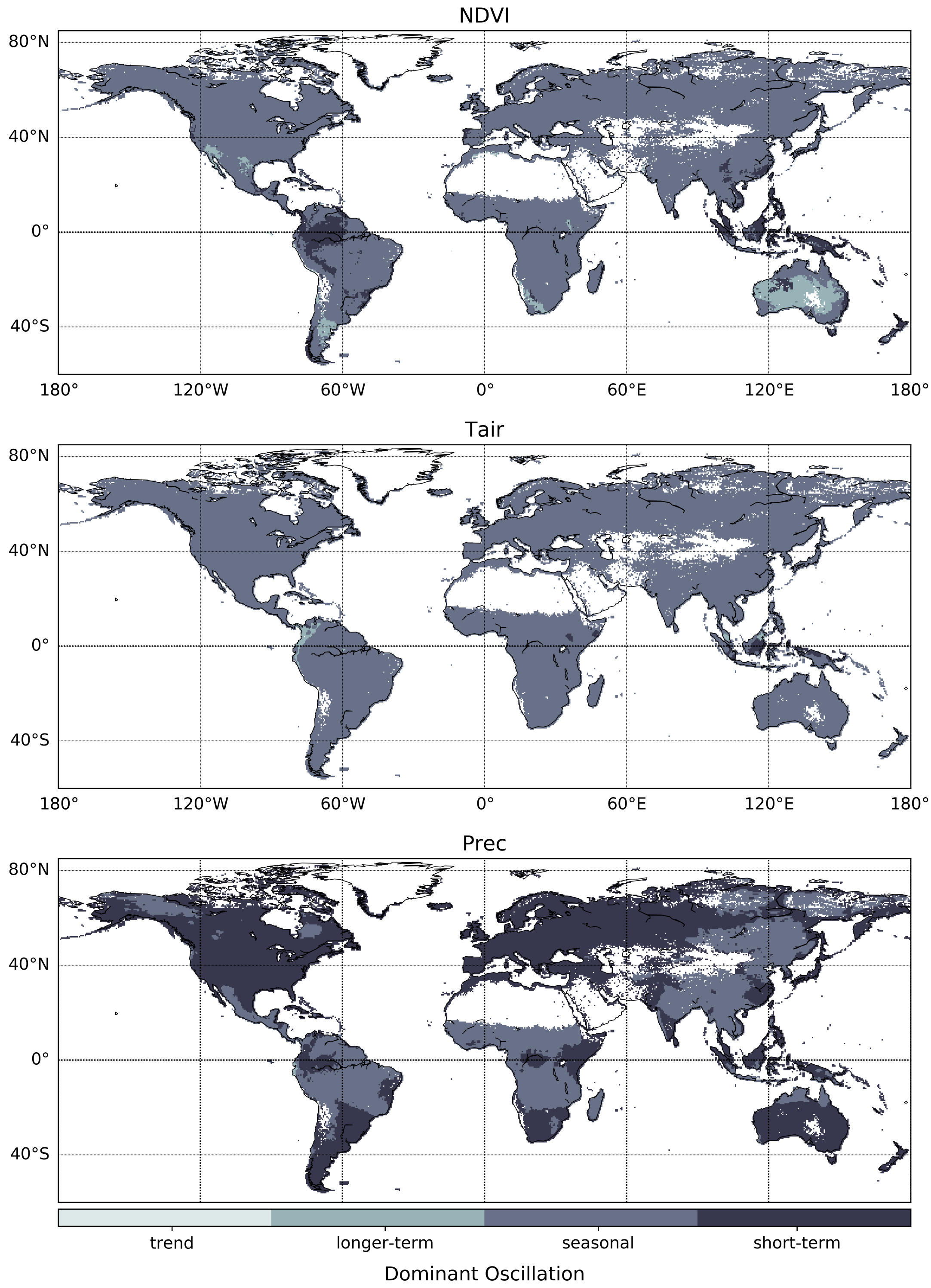


Figure S4: **Dominant Oscillation of NDVI, air temperature ($\mathbf{T}_{air}$) and precipitation (Prec) per grid cell.** Dominant scale of variability was determined from normalized, detrended and decomposed time series as the time scale containing highest relative variance (cf. Fig. 1).

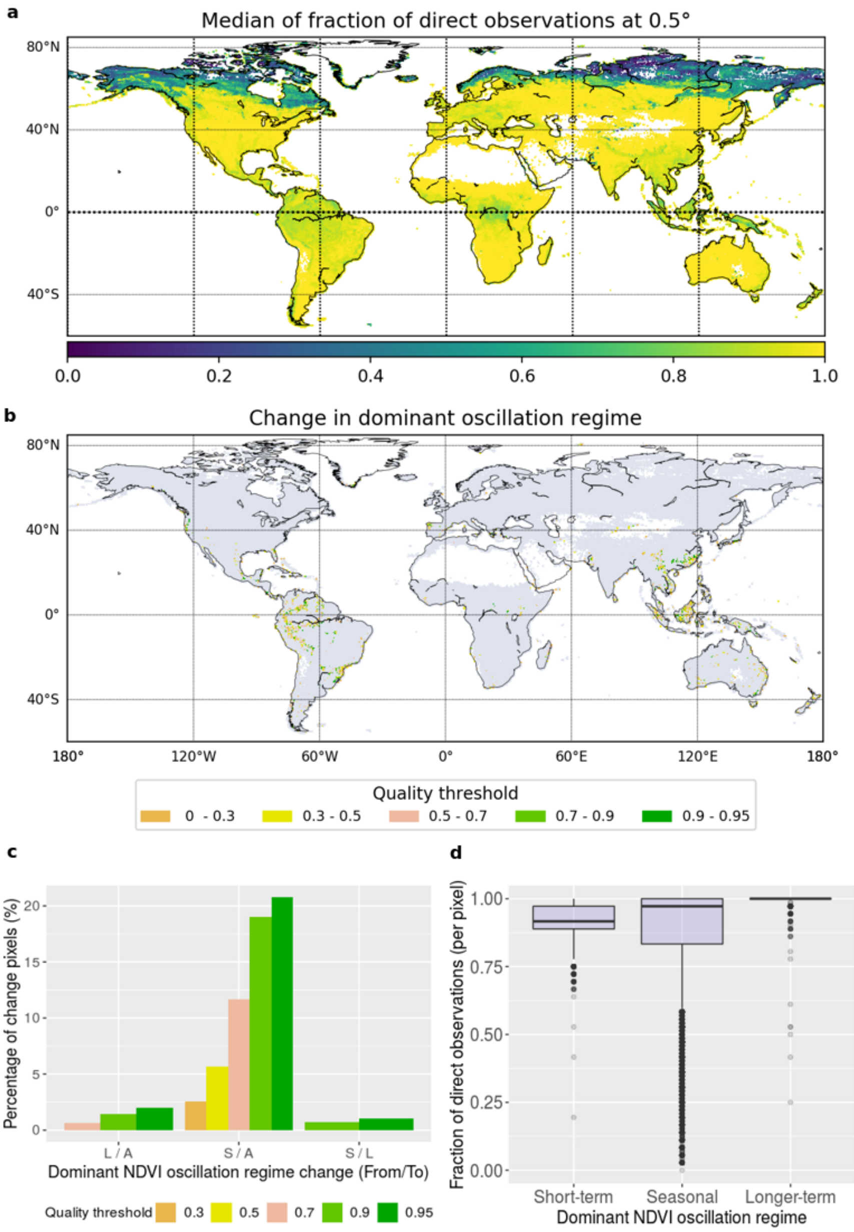


Figure S5: **Assessment of NDVI GIMMS quality flags; direct observations and effect of retrieval values. a.** Median of fraction of direct observations at 0.5° per grid cell calculated overall time period (1982–2015). Fraction of direct observations ranges from 0 to 1, and corresponds to the number of pixels with direct observation after data aggregation (from 0.083° to 0.5°). Quality flag 1 is obtained when all aggregated pixels are direct observations, 0 if none are direct observations, **b.** Pixels that change NDVI dominant oscillation class when 0.3, 0.5, 0.7, 0.9, and 0.95 quality threshold is applied (quality is defined as the fraction of pixels originating from direct observations after aggregation), **c.** Percentage of pixels with change per dominant oscillation class. S: Short–term, A: Seasonal, L: Longer–term, T: Trend, in order from / to. Categories with change $<0.05\%$ are omitted. **d.** Median fraction originating from direct observation per pixel shown as box plot per oscillation regime. Lowest percentage of direct observation is found in seasonal NDVI regimes.

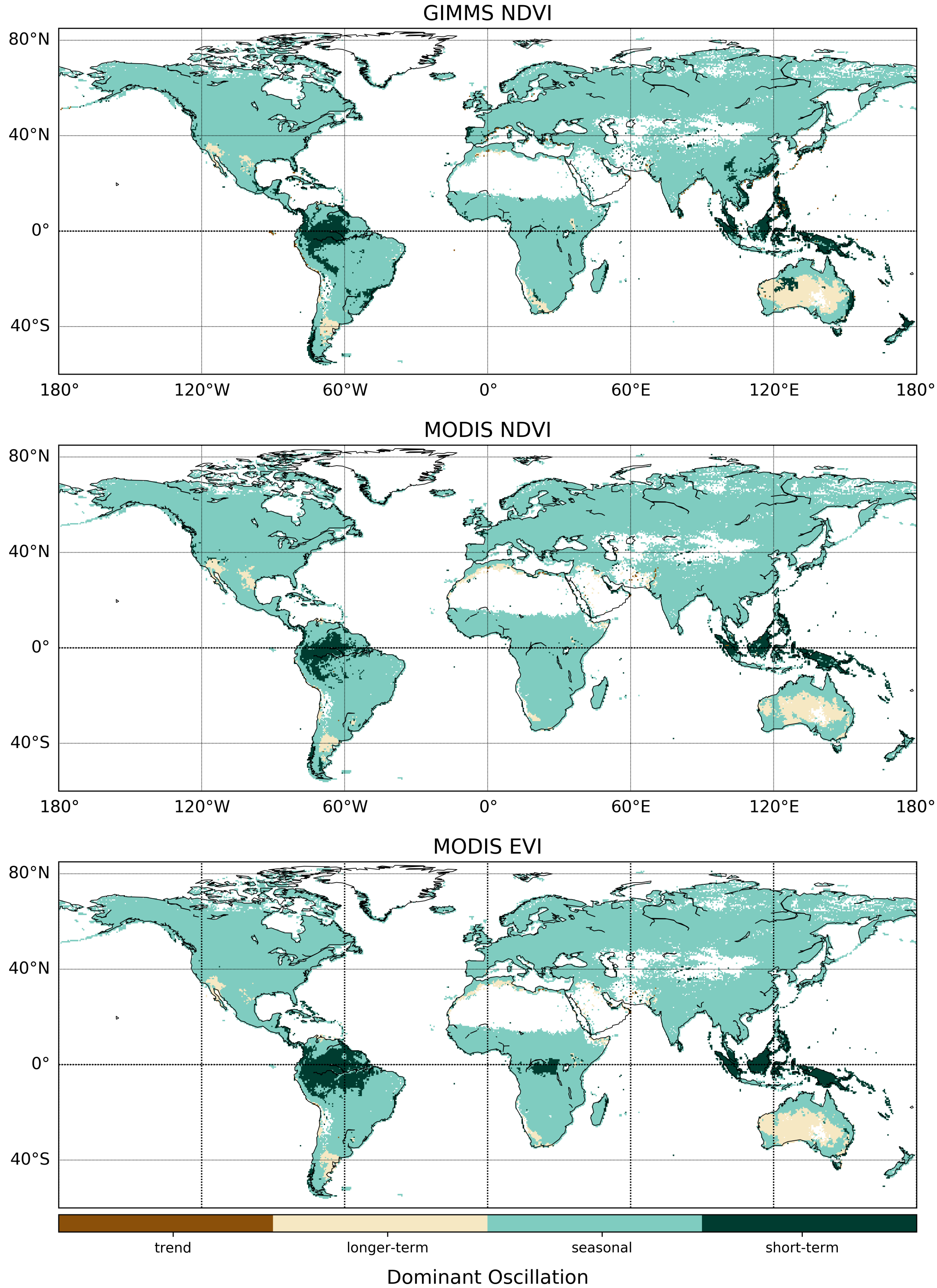


Figure S6: **Comparison of dominant oscillation classification between vegetation indices.** Dominant scale of variability for GIMMS NDVI from 1982 to 2015 (top), MODIS NDVI from 2001 to 2015 (center), and EVI MODIS from 2001 to 2015 (bottom). Dominant scale of variability was determined per pixel from normalized, detrended and Fourier–decomposed time series as the time scale containing highest relative variance.

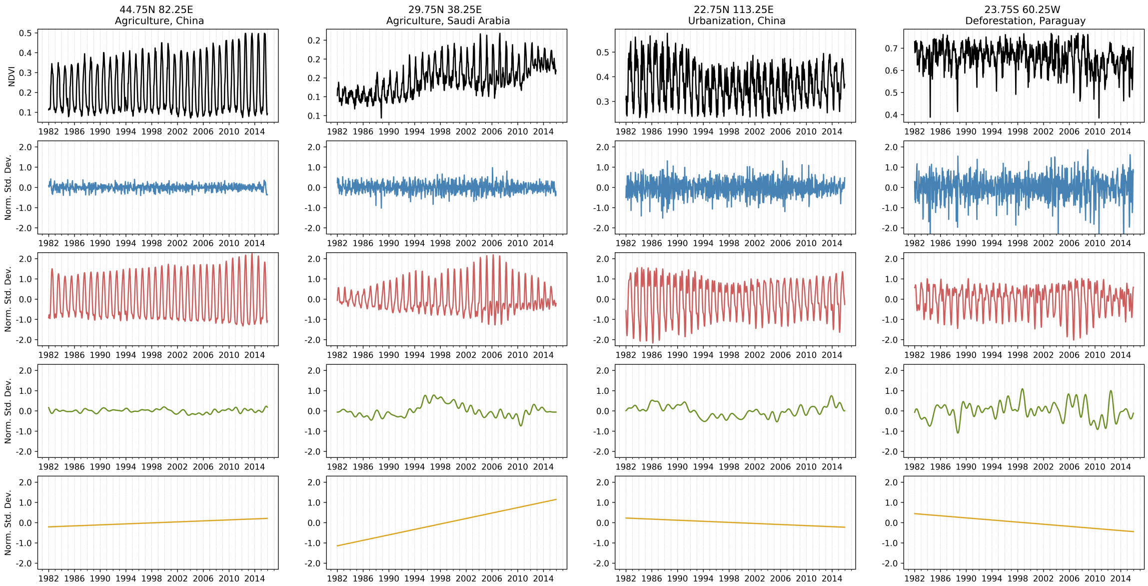


Figure S7: **Assessment of the effect of land cover change over time on decomposition results of NDVI time series** Four pixels with >25% change in vegetation type according to Song et al. (2018) are displayed (columns), representing from left to right: (i) short vegetation gain, (ii) bare ground loss, (iii) bare ground gain, and (iv) tree loss. Rows from top to bottom: integrated NDVI signal (black), short–term oscillation (blue), seasonal oscillation (red), longer–term oscillation (green), and trend (yellow). Time series were normalized and detrended before Fourier decomposition.

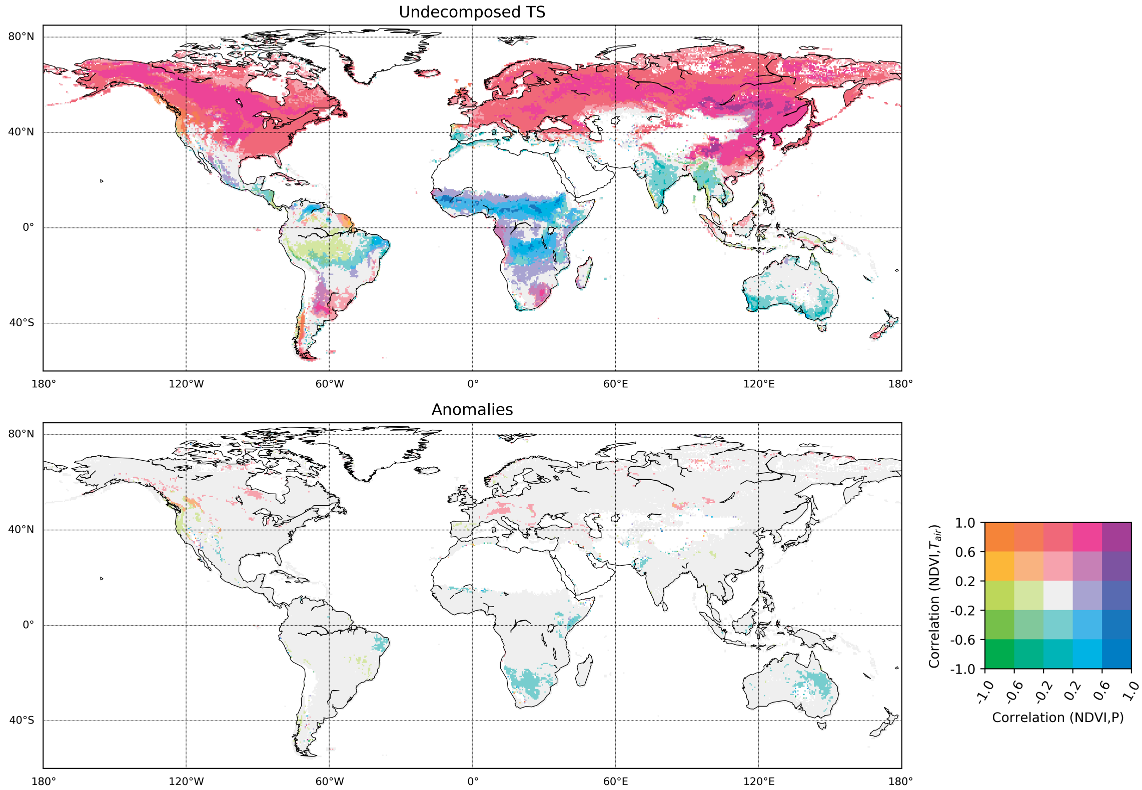


Figure S8: **Bicolor map of undecomposed time series (top) and detrended, deseasonalized anomalies (bottom).** Pearson correlation of NDVI with precipitation (Prec, legend x axis) and air temperature (T_{air}, legend y axis) is shown at each grid cell. NDVI was lagged one time step (15 days) behind precipitation to allow response time, T_{air} was correlated instantaneously. Color scale represents both correlations, binned into quantiles (e. g. purple – high positive correlation of NDVI with both T_{air} and Prec, green – high negative correlation of NDVI with both T_{air} and Prec). Data points where NDVI < 0.2 were excluded to avoid influence of inactive vegetation or non–vegetated time points.

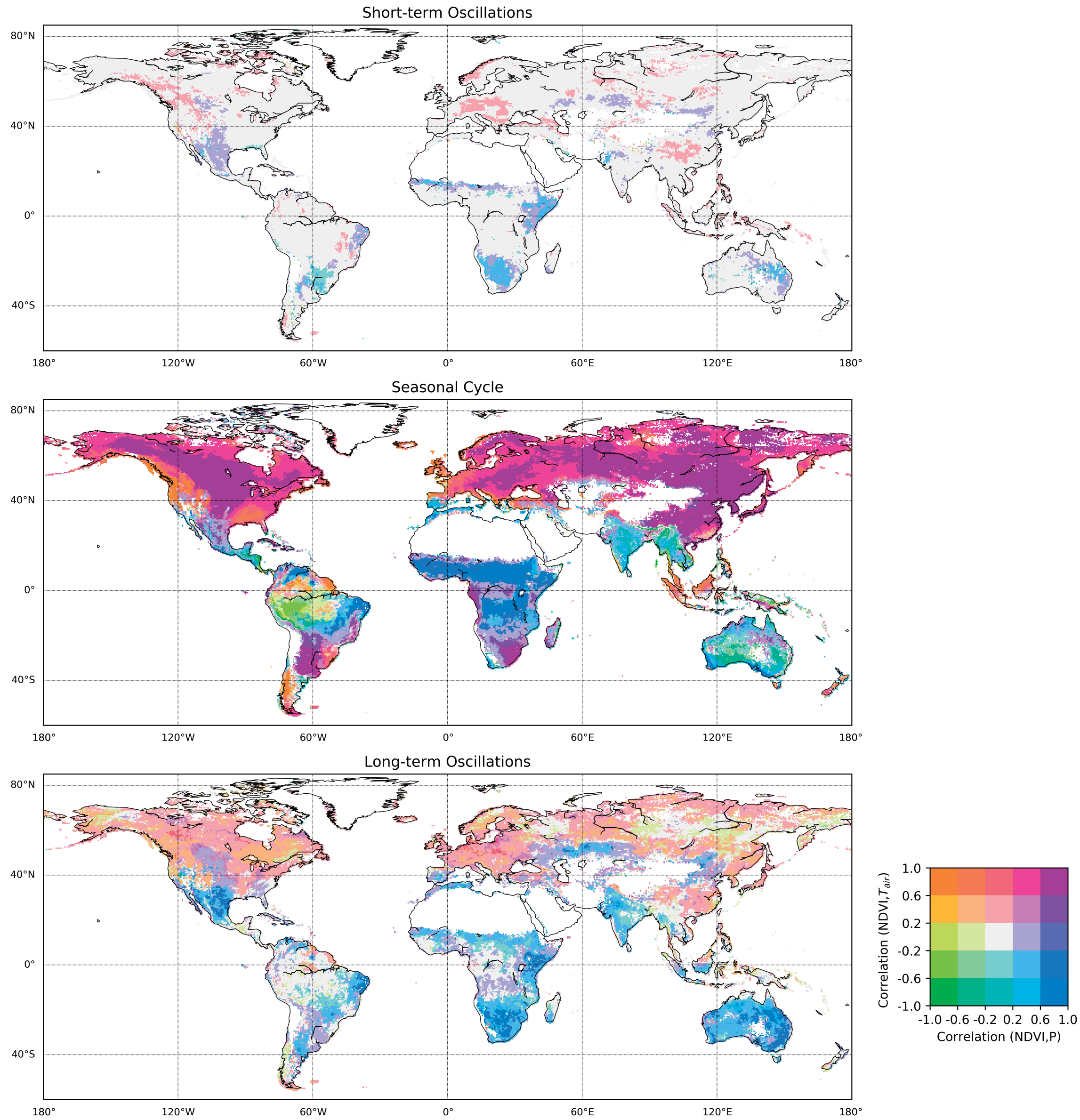


Figure S9: **Bicolor map of Spearman correlations between NDVI, air temperature (T_{air}) and precipitation (Prec).** Correlation of NDVI with T_{air} (legend y axis) and NDVI with Prec (legend x axis) were calculated between decomposed signals at each grid cell for each time scale (rows). NDVI was lagged one time step (15 days) behind precipitation to allow response time, T_{air} was correlated instantaneously. Color scale represents both correlations, binned into quantiles (e. g. purple – high positive correlation of NDVI with both T_{air} and Prec, green – high negative correlation of NDVI with both T_{air} and Prec). Data points where NDVI < 0.2 were excluded to avoid influence of inactive vegetation or non–vegetated time points. The semi–annual cycle is included in the seasonal band.

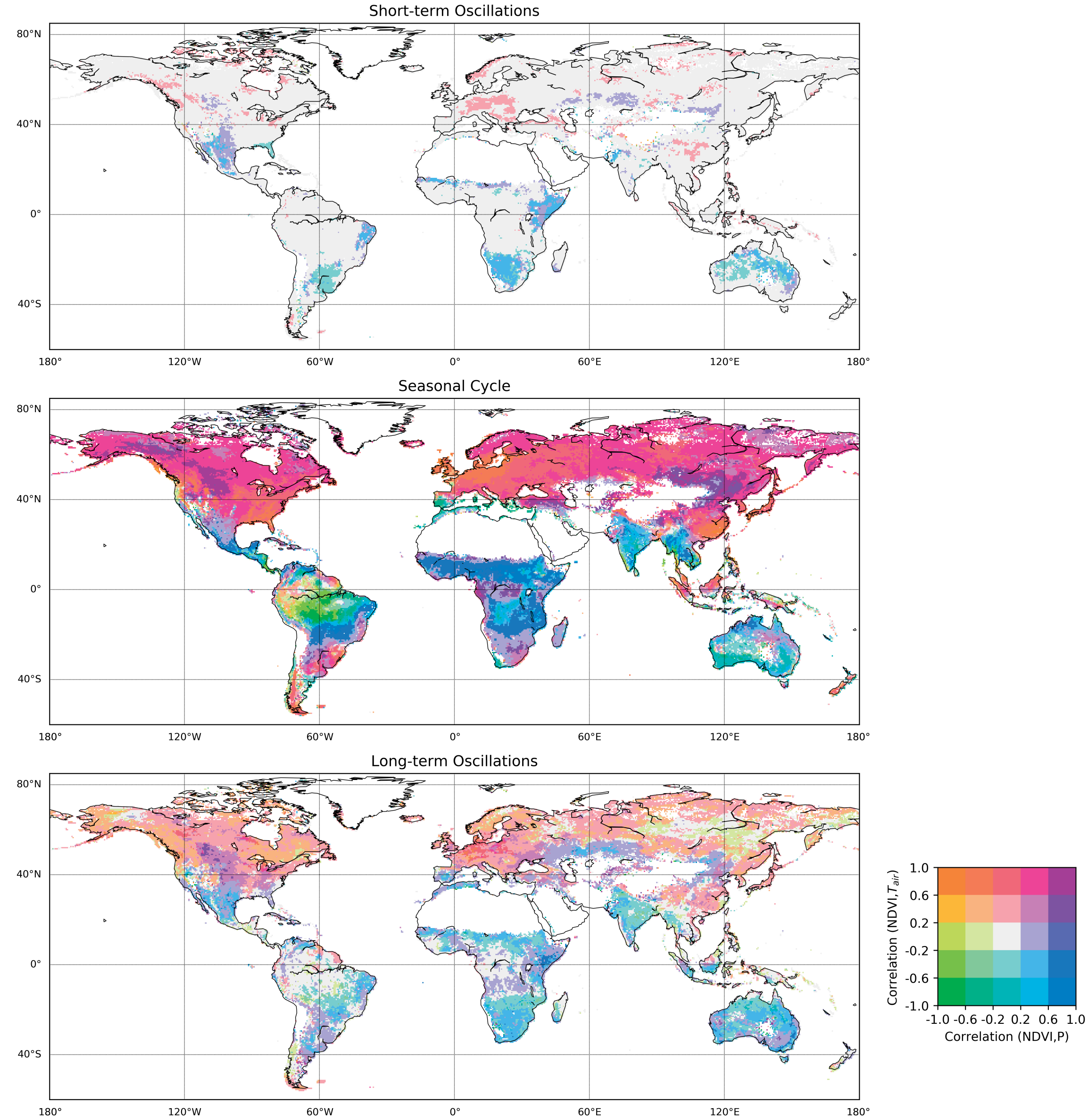


Figure S10: **Bicolor map of Partial correlations between NDVI, air temperature (T_{air}) and precipitation (Prec).** Correlation of NDVI with T_{air} (legend y axis) and NDVI with Prec (legend x axis) were calculated between decomposed signals at each grid cell for each time scale (rows). NDVI was lagged one time step (15 days) behind precipitation to allow response time, T_{air} was correlated instantaneously. Color scale represents both correlations binned into quantiles (e. g. purple – high positive correlation of NDVI with both T_{air} and Prec, green – high negative correlation of NDVI with both T_{air} and Prec). Data points where NDVI < 0.2 were excluded to avoid influence of inactive vegetation or non–vegetated time points. The semi–annual cycle is included in the seasonal band.

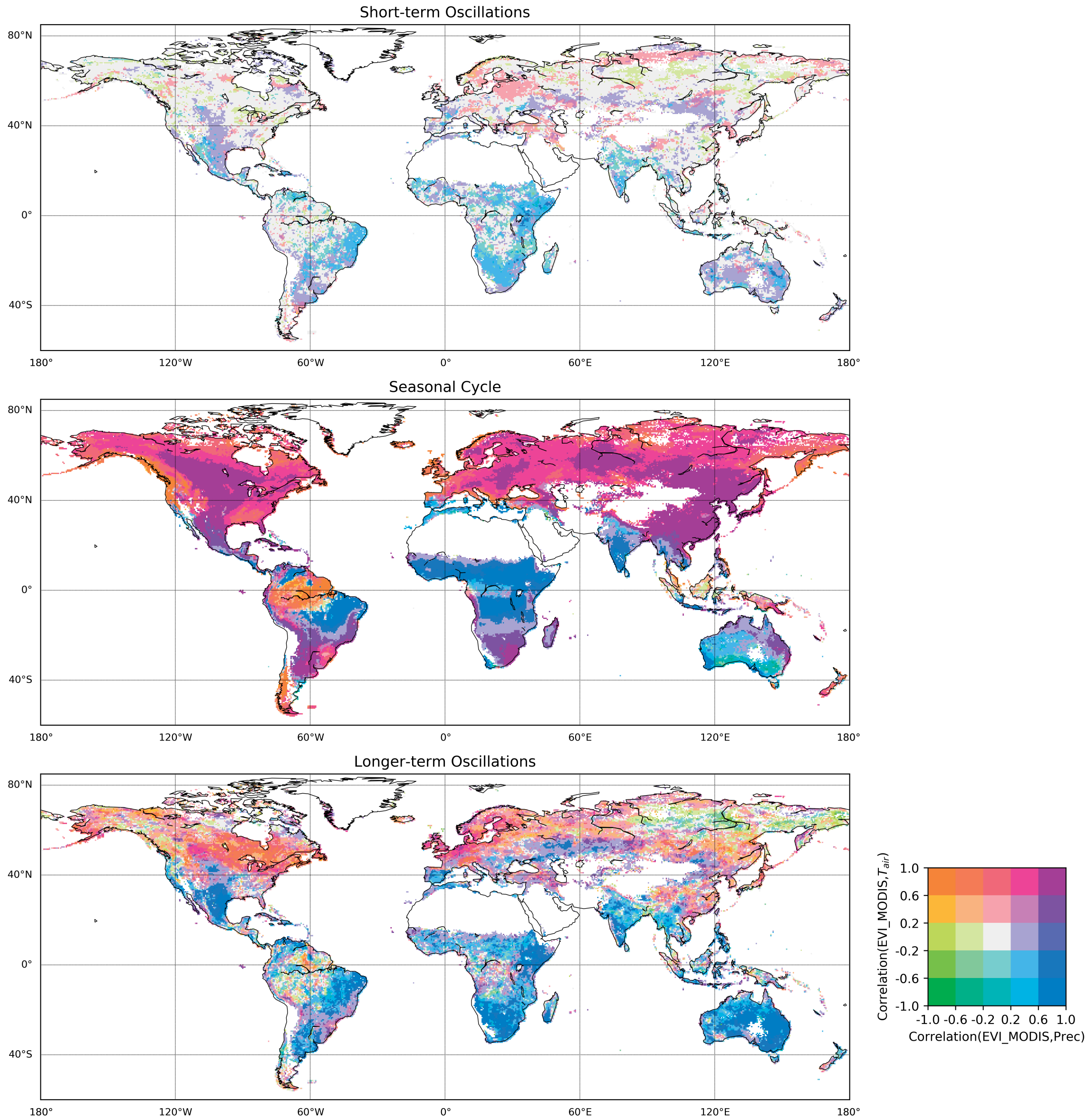


Figure S11: **Bicolor map of Pearson correlations between MODIS EVI, air temperature (T_{air}) and precipitation (Prec).** Correlation of EVI with T_{air} (legend y axis) and MODIS EVI with Prec (legend x axis) were calculated between decomposed signals at each grid cell for each time scale (rows) for the years 2007–2015. EVI was lagged one time step (15 days) behind precipitation to allow response time, T_{air} was correlated instantaneously. Color scale represents both correlations binned into quantiles (e. g. purple – high positive correlation of EVI with both T_{air} and Prec, green – high negative correlation of with both T_{air} and Prec). The semi–annual cycle is included in the seasonal band.

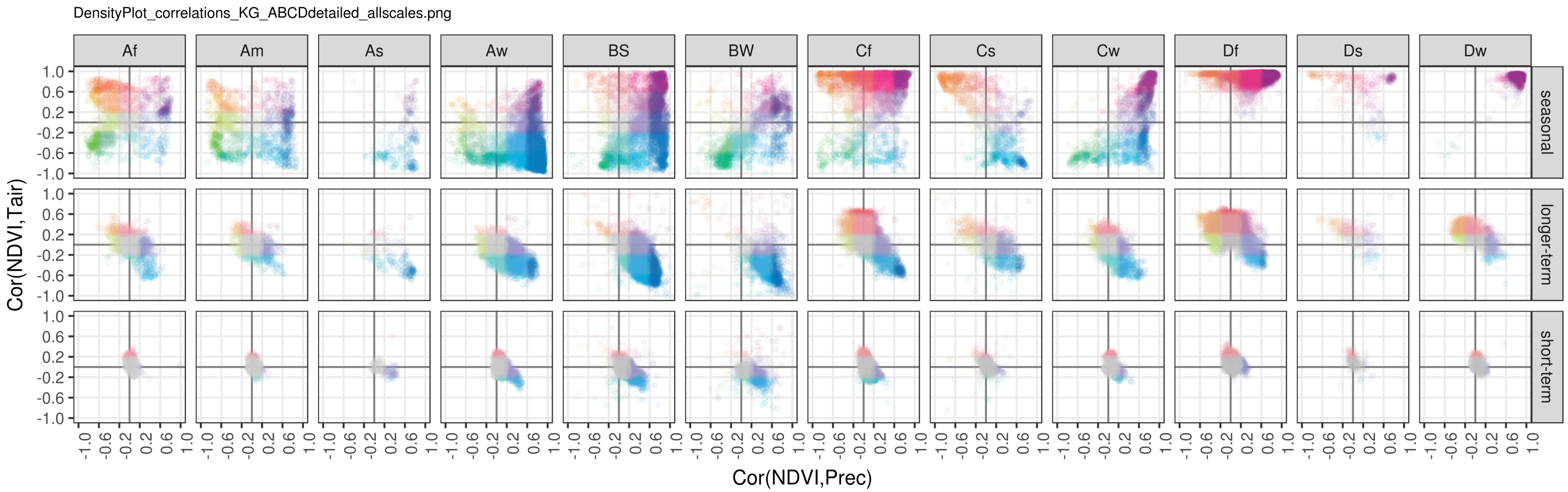


Figure S12: **Köppen-Geiger classes in "correlation space" across time scales.** Correlations of NDVI with precipitation (x axis) and air temperature (y axis) as determined for Fig. 3 were plotted for each time scale (rows) and Köppen–Geiger class (columns). Each point represents one 0.5° grid cell from the global map. Transparency was scaled with the area represented by the grid cell, colors represent both correlations in accordance with Fig. 3 binned into quantiles (e. g. purple – high positive correlation of NDVI with both T_{air} and Prec, green – high negative correlation of NDVI with both T_{air} and Prec).

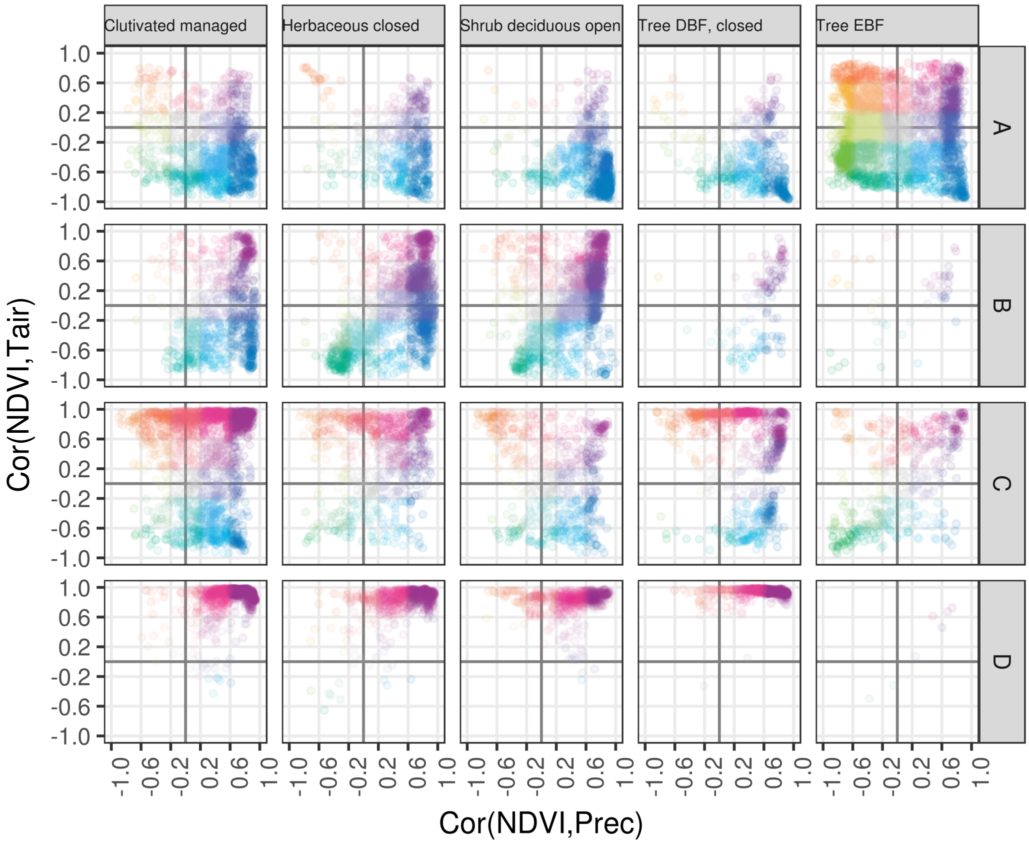


Figure S13: **Land cover classes in "correlation space" across Köppen–Geiger classes for seasonal scale.** Correlations of NDVI with precipitation (x axis) and air temperature (y axis) as determined for Fig. 3 were plotted for major land cover classes (GLC2000, columns) across Köppen–Geiger classes (rows). Each point represents one 0.5° grid cell from the global map. Transparency was scaled with the area represented by the grid cell, colors represent both correlations in accordance with Fig. 3 binned into quantiles (e. g. purple – high positive correlation of NDVI with both T_{air} and Prec, green – high negative correlation of NDVI with both T_{air} and Prec). A – equatorial, B – arid, C – warm temperate, D – snow)

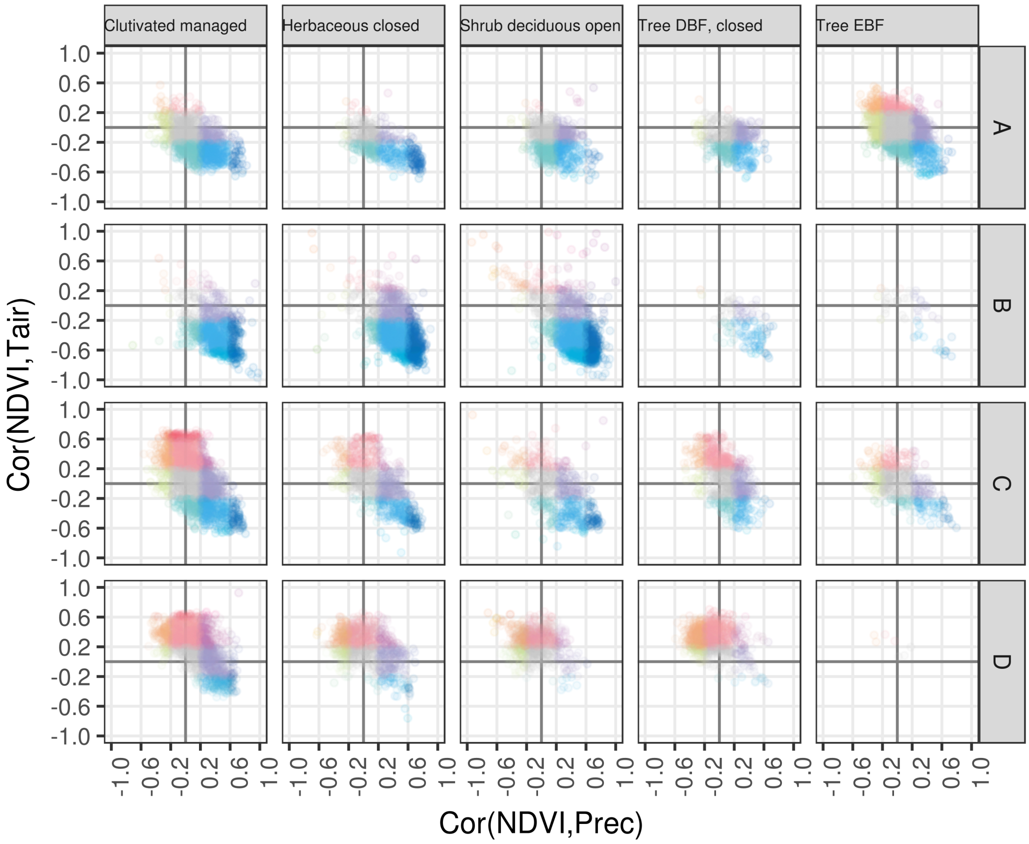


Figure S14: **Land cover classes in "correlation space" across Köppen–Geiger classes for longer–term scale.** Correlations of NDVI with precipitation (x axis) and air temperature (y axis) as determined for Fig. 3 were plotted for major land cover classes (GLC2000, columns) across Köppen–Geiger classes (rows). Each point represents one 0.5° grid cell from the global map. Transparency was scaled with the area represented by the grid cell, colors represent both correlations in accordance with Fig. 3 binned into quantiles (e. g. purple – high positive correlation of NDVI with both T_{air} and Prec, green – high negative correlation of NDVI with both T_{air} and Prec). A – equatorial, B – arid, C – warm temperate, D – snow)

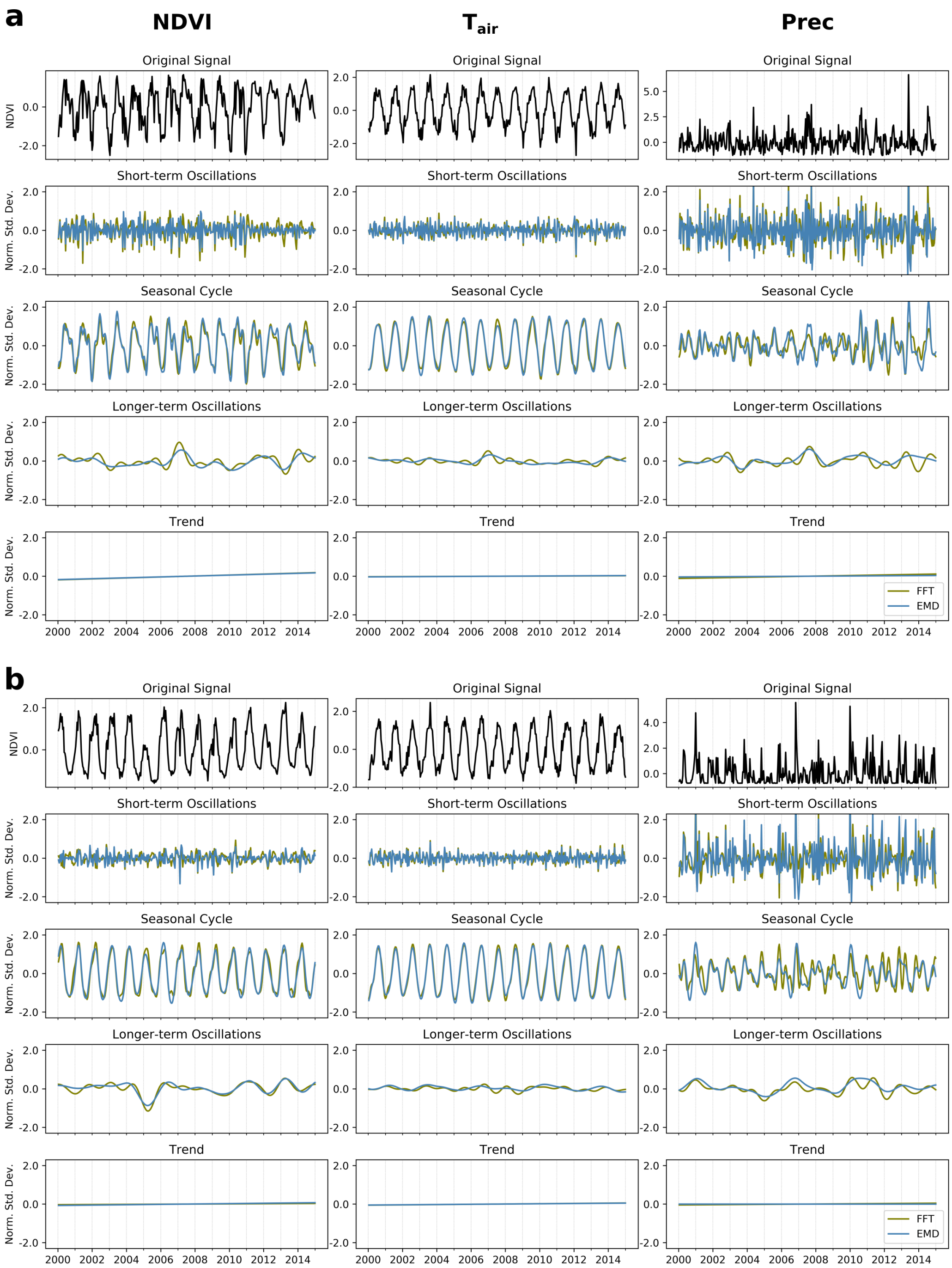


Figure S15: **Temporal comparison of Fourier transformation (FFT) and Empirical Mode Decomposition (EMD).** Time series examples for **a.** Germany (lon. 11°, lat. 51°), and **b.** southern Portugal (lon. –8°, lat. 38°) of decomposed time series of NDVI, air temperature (T_{air}) and precipitation (Prec) from 2000–2014. FFT signals are colored green, EMD signals are colored blue.

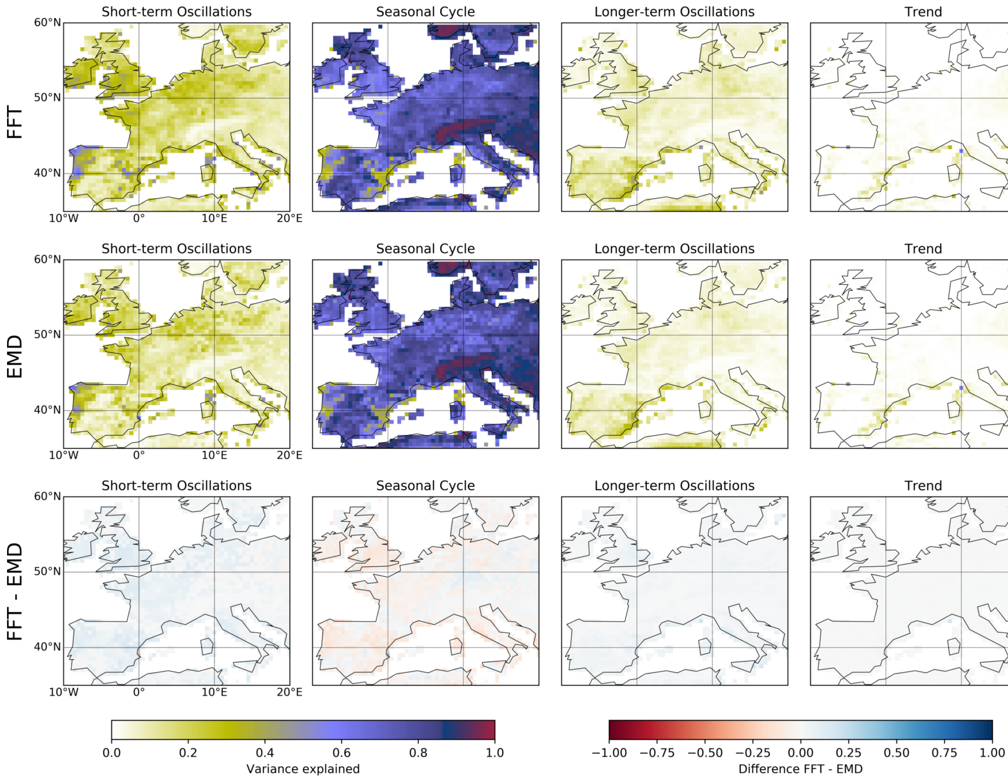


Figure S16: **Spatial comparison of Fourier transformation (FFT) and Empirical Mode Decomposition (EMD).** Comparison of variance explained per pixel as determined for NDVI time series (2000–2014) by Fourier transformation (FFT, upper row) and empirical mode decomposition (EMD), as well as their difference (lower row) over Europe.

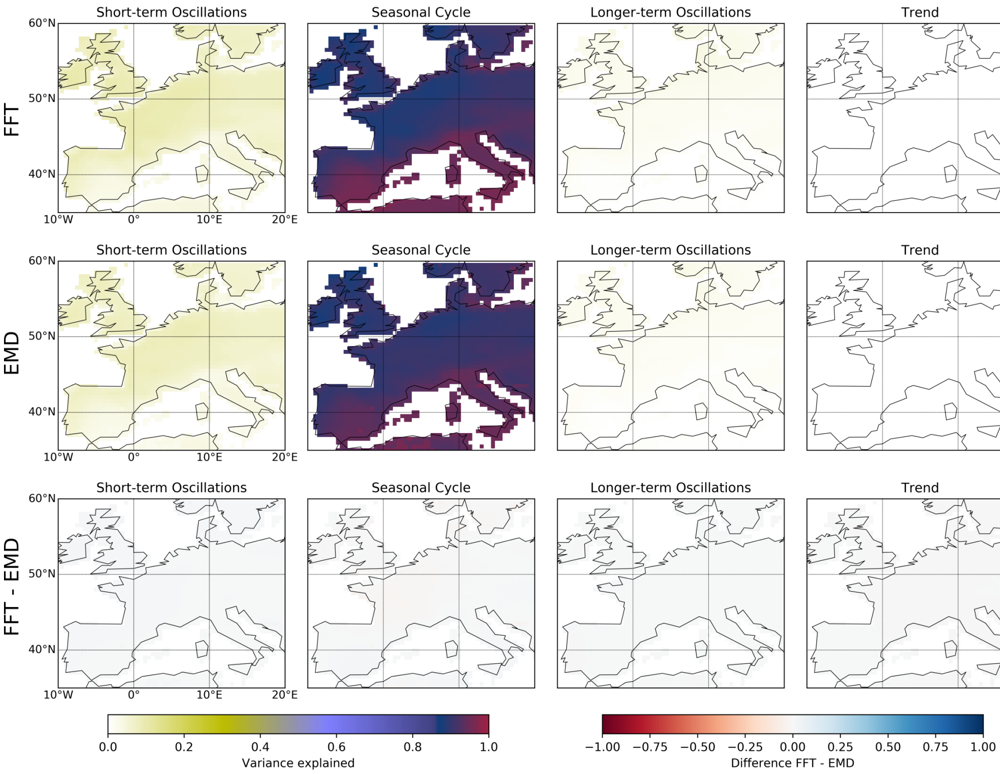


Figure S17: **Spatial comparison of Fourier transformation (FFT) and Empirical Mode Decomposition (EMD).** Comparison of variance explained per pixel as determined for air temperature time series (2000–2014) by Fourier transformation (FFT, upper row) and empirical mode decomposition (EMD), as well as their difference (lower row) over Europe.

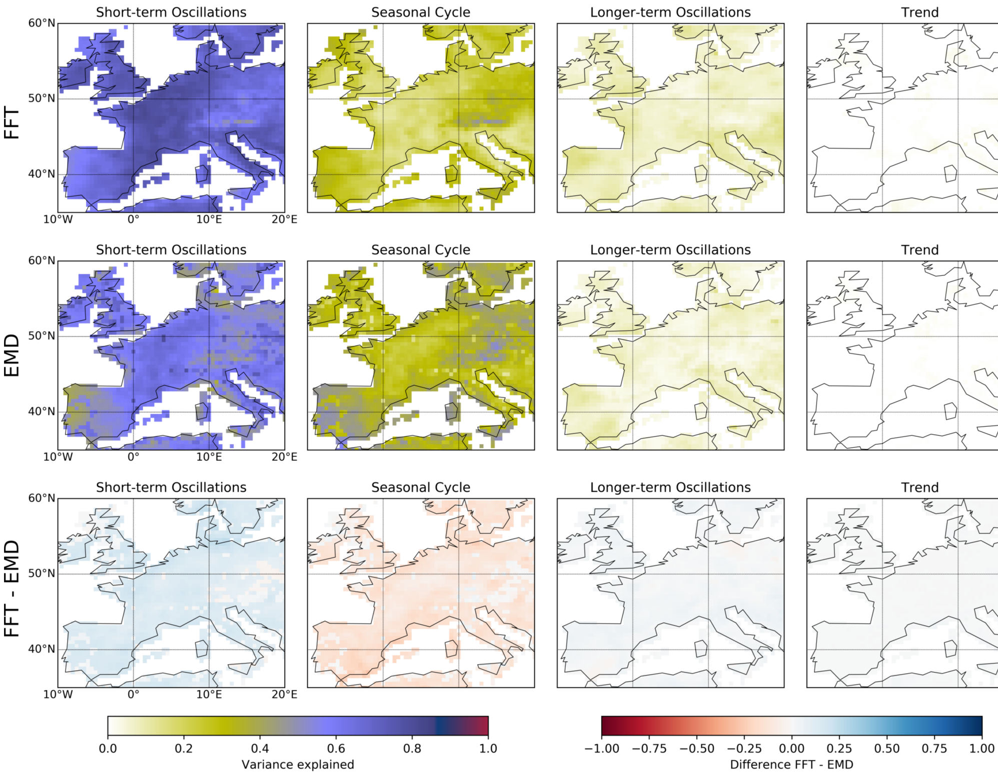


Figure S18: **Spatial comparison of Fourier transformation (FFT) and Empirical Mode Decomposition (EMD).** Comparison of variance explained per pixel as determined for precipitation time series (2000–2014) by Fourier transformation (FFT, upper row) and empirical mode decomposition (EMD), as well as their difference (lower row) over Europe.

Supplementary Tables

Table S1: Selected and excluded classes from GLC 2000.

Selected land cover classes (natural vegetation)	
Herbaceous cover closed open	Tree cover broadleaved deciduous open
Mosaic: Tree cover other natural vegetation	Tree cover broadleaved evergreen
Regularly flooded shrub and or herbaceous cover	Tree cover mixed leaf type
Shrub cover closed open deciduous	Tree cover needle leaved deciduous
Shrub cover closed open evergreen	Tree cover needle leaved evergreen
Sparse herbaceous or sparse shrub cover	Tree cover regularly flooded fresh water
Tree cover broadleaved deciduous closed	Tree cover regularly flooded saline water
Excluded land cover classes	
Artificial surfaces and associated areas	Mosaic: Cropland / Tree cover / Other natural vegetation
Bare areas	Snow and ice (natural & artificial)
Cultivated and managed areas	Tree cover burnt
Mosaic: Cropland / Shrub or grass cover	Water bodies (natural & artificial)

Table S2: Global weighted mean of decomposed oscillations and three latitudinal bands (i) extratropics northern hemisphere (above 23.5° N), (ii) tropics (23.5° N to 23.5° S) and (iii) extratropics southern hemisphere (below 23.5° S). The mean weights are based on pixel area.

Variable	**Region**	**Short-term**	**Seasonal**	**Longer-term**	**Trend**
NDVI	Global	0.18	0.71	0.09	0.02
NDVI	Above 23.5° N	0.1	0.84	0.05	0.01
NDVI	Tropics	0.27	0.59	0.11	0.02
NDVI	Below 23.5° S	0.25	0.46	0.25	0.03
Tair	Global	0.11	0.83	0.04	0.01
Tair	Above 23.5° N	0.05	0.94	0.01	0
Tair	Tropics	0.21	0.68	0.09	0.02
Tair	Below 23.5° S	0.08	0.9	0.02	0
Prec	Global	0.52	0.41	0.06	0
Prec	Above 23.5° N	0.57	0.36	0.06	0
Prec	Tropics	0.42	0.51	0.06	0
Prec	Below 23.5° S	0.68	0.24	0.09	0

Table S3: Summary statistics of total area assessed and percentage of dominant NDVI oscillations by Köppen–Geiger, vegetated land cover classes and dominant oscillations of climatic variables. A: Annual, L: Longer–term, S:Short–term, T: Trend. Values of T are solely presented for area calculations.

	DOMINANT OSCILLATION OF NDVI												
	Area assessed (km2)				Overall percentage (%)			Percentage by class (% rows)			Percentage by dominant NDVI (% cols.)		
Köppen-Geiger classes	**S**	**A**	**L**	**T**	**S**	**A**	**L**	**S**	**A**	**L**	**S**	**A**	**L**
ColdTemp	20666.8	23707941.1	0.0	10321.0	0.0	31.2	0.0	0.1	99.9	0.0	0.3	37.1	0.0
WarmTemp	1531328.9	11281304.1	336226.0	78194.2	2.0	14.9	0.4	11.6	85.3	2.5	22.4	17.6	6.9
Arid	775669.8	13231895.0	4534532.5	79587.2	1.0	17.4	6.0	4.2	71.1	24.4	11.3	20.7	92.5
Equatorial	4520718.3	15702426.7	30631.2	30043.6	6.0	20.7	0.0	22.3	77.4	0.2	66.0	24.6	0.6
Vegetated land cover classes	**S**	**A**	**L**	**T**	**S**	**A**	**L**	**S**	**A**	**L**	**S**	**A**	**L**
Herbaceous cover closed open	514350.8	9193800.1	1255570.0	19281.0	0.7	12.1	1.7	4.7	83.7	11.4	7.5	14.4	25.6
Mosaic: Tree cover other natural vegetation	138263.4	2050350.8	5415.6	5717.1	0.2	2.7	0.0	6.3	93.2	0.2	2.0	3.2	0.1
Regularly flooded shrub and or herbaceous cover	94256.8	1425427.4	51408.3	2849.5	0.1	1.9	0.1	6.0	90.6	3.3	1.4	2.2	1.0
Shrub cover closed open deciduous	208024.5	9421620.4	966549.2	33200.1	0.3	12.4	1.3	2.0	88.6	9.1	3.0	14.7	19.7
Shrub cover closed open evergreen	210446.9	1688229.8	41985.2	7422.2	0.3	2.2	0.1	10.8	86.7	2.2	3.1	2.6	0.9
Sparse herbaceous or sparse shrub cover	637336.7	7667753.4	2404181.6	61753.6	0.8	10.1	3.2	5.9	71.2	22.3	9.3	12.0	49.1
Tree cover broadleaved deciduous closed	140333.1	6241346.3	17724.3	19055.9	0.2	8.2	0.0	2.2	97.2	0.3	2.0	9.8	0.4
Tree cover broadleaved deciduous open	97232.9	3473408.2	81498.9	0.0	0.1	4.6	0.1	2.7	95.1	2.2	1.4	5.4	1.7
Tree cover broadleaved evergreen	4107891.8	7421853.2	17685.7	26460.1	5.4	9.8	0.0	35.5	64.1	0.2	60.0	11.6	0.4
Tree cover mixed leaf type	37583.7	3049469.0	2419.7	9124.3	0.0	4.0	0.0	1.2	98.4	0.1	0.5	4.8	0.0
Tree cover needle leaved deciduous	0.0	3427749.0	0.0	0.0	0.0	4.5	0.0	0.0	100.0	0.0	0.0	5.4	0.0
Tree cover needle leaved evergreen	389722.8	8498764.0	56951.3	13282.3	0.5	11.2	0.1	4.4	94.9	0.6	5.7	13.3	1.2
Tree cover regularly flooded fresh water	257565.2	303364.8	0.0	0.0	0.3	0.4	0.0	45.9	54.1	0.0	3.8	0.5	0.0
Tree cover regularly flooded saline water	15375.3	60430.6	0.0	0.0	0.0	0.1	0.0	20.3	79.7	0.0	0.2	0.1	0.0
Dominant oscillations of Temperature and precipitation*	**S**	**A**	**L**	**T**	**S**	**A**	**L**	**S**	**A**	**L**	**S**	**A**	**L**
A.A.	2505858.1	26763039.4	0.0	0.0	3.3	35.3	0.0	8.6	91.4	0.0	36.6	41.9	0.0
A.S.	3767937.1	36854191.1	4901389.7	198146.0	5.0	48.6	6.5	8.2	80.6	10.7	55.0	57.7	100.0
S.A.	0.0	33532.8	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.1	0.0
S.S.	338242.5	196420.7	0.0	0.0	0.4	0.3	0.0	63.3	36.7	0.0	4.9	0.3	0.0
L.S.	236346.1	0.0	0.0	0.0	0.3	0.0	0.0	100.0	0.0	0.0	3.5	0.0	0.0
L.A.	0.0	76383.0	0.0	0.0	0.0	0.1	0.0	0.0	100.0	0.0	0.0	0.1	0.0
Percentage area assessed (%)	9.03	84.25	6.46	0.26									
Total area assessed (km^2)	75871486.5												

*A: Annual, S: Short-term oscillation, L: Longer-term oscillation, T: Trend

Table S4: Spatial association between co–oscillation regimes and Köppen–Geiger or Global Land Cover (GLC2000), respectively. c = complementarity, h = homogeneity, m = number of classes , V = V–measure.

	Co-oscillations regime (11 classes)			
Static maps	**m**	**h**	**c**	**V**
Köppen–Geiger	4	0.19	0.16	0.17
Global land cover	9	0.16	0.09	0.11

References

Song, X.-P., Hansen, M. C., Stehman, S. V., Potapov, P. V., Tyukavina, A., Vermote, E. F., & Townshend, J. R. (2018, August). Global land change from 1982 to 2016. *Nature*, *560*, 639–643. doi: 10.1038/s41586-018-0411-9

Supplementary Notebook

Data Analysis

Towards a global understanding of vegetation–climate dynamics at multiple time scales.

Linscheid, Estupinan-Suarez, et al., Biogeoscience [https://doi.org/10.5194/bg-2019-321 (https://doi.org/10.5194/bg-2019-321)](https://doi.org/10.5194/bg-2019-321)

correspondence to: nlinsch@bgc-jena.mpg.de, lestup@bgc-jena.mpg.de

Time series decomposition by Fourier, analysis of variance per time scale, dominant time scales, and difference in sign of vegetation-climate correlation across time scales

- This notebook exemplifies the main analysis in the paper "Towards a global understanding of vegetation–climate dynamics at multiple time scales"
- The notebook is written in Julia 0.6
- "#" comments in the code are intended to explain specific aspects of the coding
- "##" comments in the code are intended to describe datasets or objects for clarification
- New steps in workflows are introduced with bold headers
- Map plots are used to illustrate the outcomes. They graphically differ from figures in the paper (e.g. colormaps, axis) which were mainly produced in Python.
- Datasets pre-processing is not included on this notebook but introduced briefly. Access to the code or data is available by the correspondance authors.

June 2019, Max Planck Insitute of Biogeochemistry, Jena, Germany

Load required packages

In [1]:
```
# for operating the Earth system data lab (ESDL v0.4.15)
using ESDL, ESDLPlots

# for tracking process progress
using ProgressMeter

# for loading aggregated data from home directory into ESDL (BGIData v0.1.3)
using BGIData

# for working with missing values
using Missings

# for reading CSV files
using CSV
```

```
INFO: Interact.jl: using new nbwidgetsextension protocol
```

In [2]:
```
# for heatmaps
using Plots
gr(html_output_format=:png)
```

Out[2]:
```
Plots.GRBackend()
```

In [3]:
```
missing_to_nan!(x::Array{Union{Missing, T}}) where T = map!(y -> y === missing
? T(NaN) : y, x, x)
missing_to_nan(x::Array{Union{Missing, T}}) where T = map(y -> y === missing ?
T(NaN) : y, x)
missing_to_nan(x::Array{T}) where T = x
missing_to_nan!(x::Array{T}) where T = x
missing_to_zero!(x::Array{Union{Missing, T}}) where T = map!(y -> y === missin
g ? T(0) : y, x, x)
missing_to_zero(x::Array{Union{Missing, T}}) where T = map(y -> y === missing
? T(0) : y, x)
```

Out[3]:
```
missing_to_zero (generic function with 1 method)
```

Data and preprocessing

Load 15-daily data (NDVI, Tair, Prec)

In [4]:
```
path_ndvi, path_tair, path_prec, path_oce_des = readlines("datapaths.txt");
```

In [5]:
```
## Data: NDVI - Source: GIMMS v3.1 (Pinzon and Tucker, 2014)
## pre-processing: data was previously resampled from 0.083°
## to 0.5° by averaging

ds_ndvi = bgi_from_dir(path_ndvi, "NDVI")
```

Out[5]: **NDVI** *NDVI* Resolution: 0.5deg; Years 1982.0-2016.0

In [6]:
```
## Data: air temperature - Source: ERA Interim v4 (Dee et al., 2011)
## pre-processing: data was previously aggregateted to 15-daily time steps
## by averaging

ds_temp = bgi_from_dir(path_tair, "Tair")
```

Out[6]: **Tair** *Tair* Resolution: 0.5deg; Years 1982.0-2016.0

In [7]:
```
## Data: precipitation - Source: MSWEP (Beck, Wood, et al., 2019)
## pre-processing: data was previously aggregateted to 15-daily time steps
## by summation, and resampled from 0.083° to 0.5° by averaging

ds_mswep = bgi_from_dir(path_prec, "prec")
```

Out[7]: **prec** *prec* Resolution: 0.5deg; Years 1982.0-2016.0

Load cubes

In [8]:
```
# define time period

t = (Date(1982,1,1),Date(2015,12,31))
```

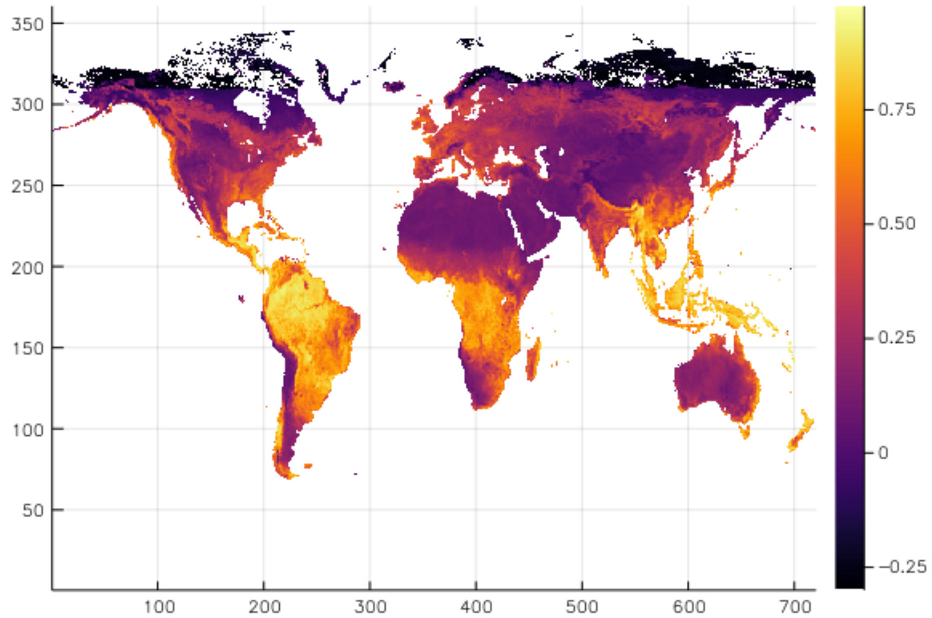
Out[8]:
```
(1982-01-01, 2015-12-31)
```

In [9]:
```
# get cubes independently for each variable

c_vegidx = getCubeData(ds_ndvi, time=t)
c_prec = getCubeData(ds_mswep, time=t)
c_temp = getCubeData(ds_temp, time=t)
```

Out[9]:
```
BGIData variable Tair with the following dimensions
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Time                Axis with 816 Elements from 1982-01-01 to 2015-12-15
Total size: 1.77 GB
```

In [10]:
```
# plotting first time step of NDVI data cube "c_vegidx"

heatmap(missing_to_nan(c_vegidx[:,:,1])'[end:-1:1,:])
```

Out[10]:



Gapfilling

In [11]:
```
# here only for NDVI, the others do not have gaps

c_fill_vegidx = gapFillMSC(c_vegidx)
```

```
Progress: 100%|████████████████████████████████████████| Time: 0:01:49:33
```

Out[11]:
```
Memory mapped cube with the following dimensions
Time                Axis with 816 Elements from 1982-01-15T00:00:00 to 2015-12
-31T00:00:00
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 1.77 GB
```

Normalization

(Z-scoring per pixel)

In [12]:
```
c_norm_vegidx = normalizeTS(c_fill_vegidx)
c_norm_prec = normalizeTS(c_prec)
c_norm_temp = normalizeTS(c_temp)
```

```
Progress: 100%|██████████████████████████████████████████| Time: 0:00:25
Progress: 100%|██████████████████████████████████████████| Time: 0:01:470:19
Progress: 100%|██████████████████████████████████████████| Time: 0:01:46
```

Out[12]:
```
Memory mapped cube with the following dimensions
Time                Axis with 816 Elements from 1982-01-01 to 2015-12-15
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 1.77 GB
```

FFT decomposition

In [13]:
```
## Fast Fourier decomposition

c_fft_vegidx = filterTSFFT(c_norm_vegidx)
c_fft_prec = filterTSFFT(c_norm_prec)
c_fft_temp = filterTSFFT(c_norm_temp)
```

```
Progress: 100%|██████████████████████████████████████████| Time: 0:01:20:19
Progress: 100%|██████████████████████████████████████████| Time: 0:01:21
Progress: 100%|██████████████████████████████████████████| Time: 0:01:22
```

Out[13]:
```
Memory mapped cube with the following dimensions
Time                Axis with 816 Elements from 1982-01-01 to 2015-12-15
Scale               Axis with elements: Trend Long-Term Variability Annual Cyc
le Fast Oscillations
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 7.09 GB
```

Percent missing values in NDVI

In [14]:
```
using Missings
"""
Calculate proportion valid (not 'missing' or 'NaN') values along a vector

r = rand(10)
r[10] = NaN
percentvalid(r) # => 0.9

"""
function percentvalid(xin) # where xin is a timeseries
    a=count(.!ismissing.(xin) .& .!isnan.(xin)) # count number of valid values
    b=length(xin) # length of TS
    return a/b # fraction of valid values
end
```

Out[14]:
```
percentvalid
```

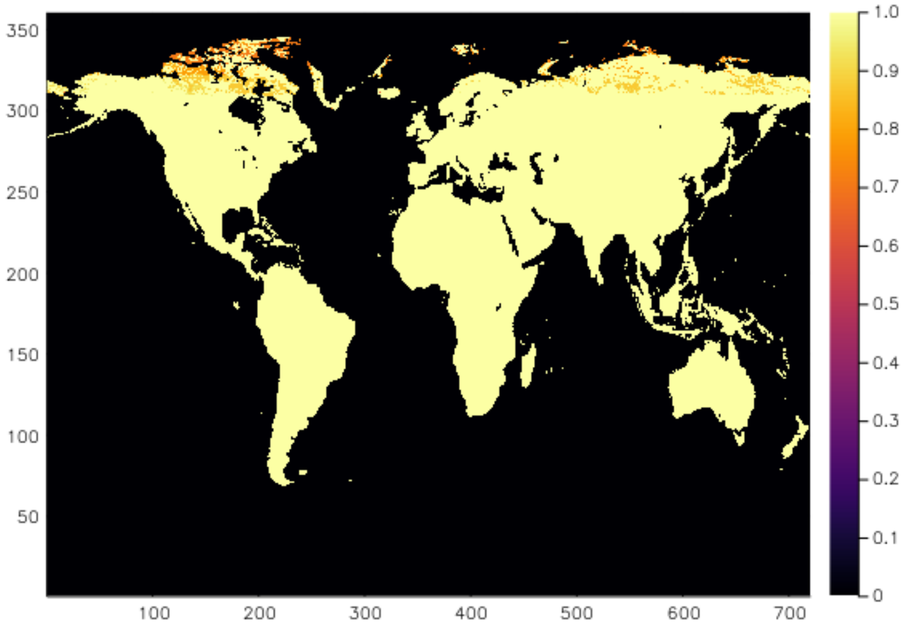
In [15]:
```
c_gapmask = mapslices(percentvalid, c_vegidx, "Time")
```

```
Progress: 100%|██████████████████████████████████████████| Time: 0:01:47
```

Out[15]:
```
In-Memory data cube with the following dimensions
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 2.22 MB
```

In [16]:
```
# map of the fraction of valid values

heatmap(missing_to_nan(c_gapmask[:,:])'[end:-1:1,:])
```

Out[16]:



Masks

In [17]:
```
# mask out gapfilled values again for downstream analyses

mask_gaps = permutedims(c_vegidx[:,:,:],[3,1,2]) .== c_fill_vegidx[:,:,:];
```

In [18]:
```
# ndvi cutoff mask at 0.2 (remove time points with non-active vegetation)

mask_ndvi = map(i-> isnan.(i) ? false : i.<0.2 ? false : true,
    c_fill_vegidx[:,:,:]);
```

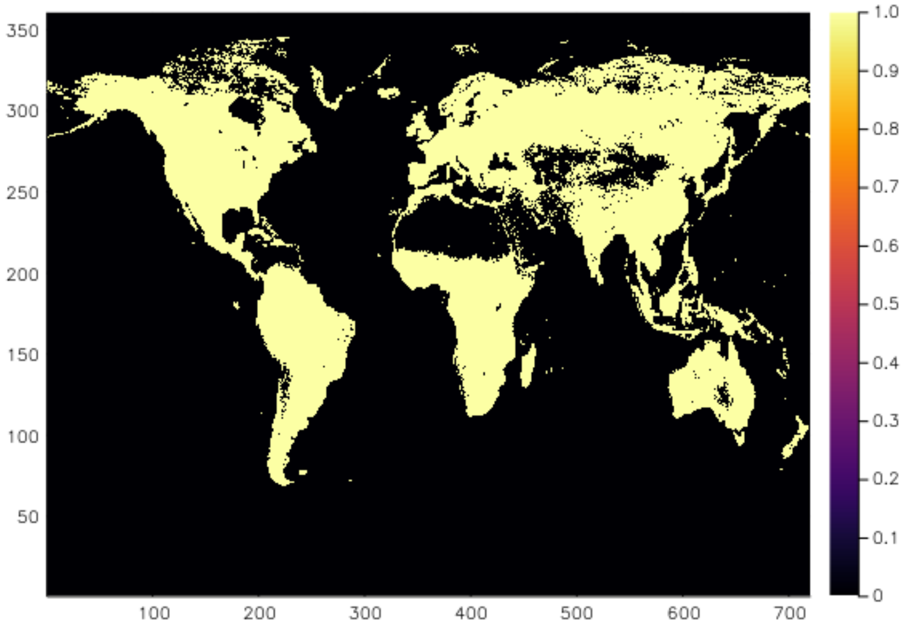
In [19]:
```
# combine both masks in one

mask_combined = mask_gaps .& mask_ndvi;
```

In [20]:
```
# mask for ocean and desert

oce_des_df = CSV.read(path_oce_des, header = false);
```

In [25]:
```
# converting ocean desert mask data frame into a matrix

oce_des = missing_to_nan(oce_des_df[:,:] |> Matrix |> permutedims)
oce_des_mask = convert(Array{Float64,2}, map(x -> x==0 ? NaN : x, oce_des));
```

In [26]:
```
# map ocean & dessert mask

heatmap(oce_des'[end:-1:1,:])
```

Out[26]:



Make into a cube

In [21]:
```
# create a cube for gapfilled mask

cubemask_gapfilled = ESDL.CubeMem(
    CubeAxis[c_fill_vegidx.axes[1], c_fill_vegidx.axes[2], c_fill_vegidx.axes
[3]],
    map(i->i ? 1.0 : NaN, mask_gaps),
    map(i->i ? ESDL.Mask.VALID : ESDL.Mask.MISSING, mask_gaps))
```

Out[21]:
```
In-Memory data cube with the following dimensions
Time                Axis with 816 Elements from 1982-01-15T00:00:00 to 2015-12
-31T00:00:00
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 1.77 GB
```

In [22]:
```
# create a cube for combined mask

cubemask = ESDL.CubeMem(
    CubeAxis[c_fill_vegidx.axes[1], c_fill_vegidx.axes[2], c_fill_vegidx.axes
[3]],
    map(i->i ? 1.0 : NaN, mask_combined),
    map(i->i ? ESDL.Mask.VALID : ESDL.Mask.MISSING, mask_combined))
```

Out[22]:
```
In-Memory data cube with the following dimensions
Time                Axis with 816 Elements from 1982-01-15T00:00:00 to 2015-12
-31T00:00:00
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 1.77 GB
```

In [23]:
```
# calculate number of valid values over time after applying all filtering

c_percmask = mapslices(percentvalid, cubemask, "Time")
```

```
Progress: 100%|██████████████████████████████████████████| Time: 0:00:06
```

Out[23]:
```
In-Memory data cube with the following dimensions
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 2.22 MB
```

In [24]:
```
heatmap(c_percmask[:,:]'[end:-1:1,:])
```

Out[24]:

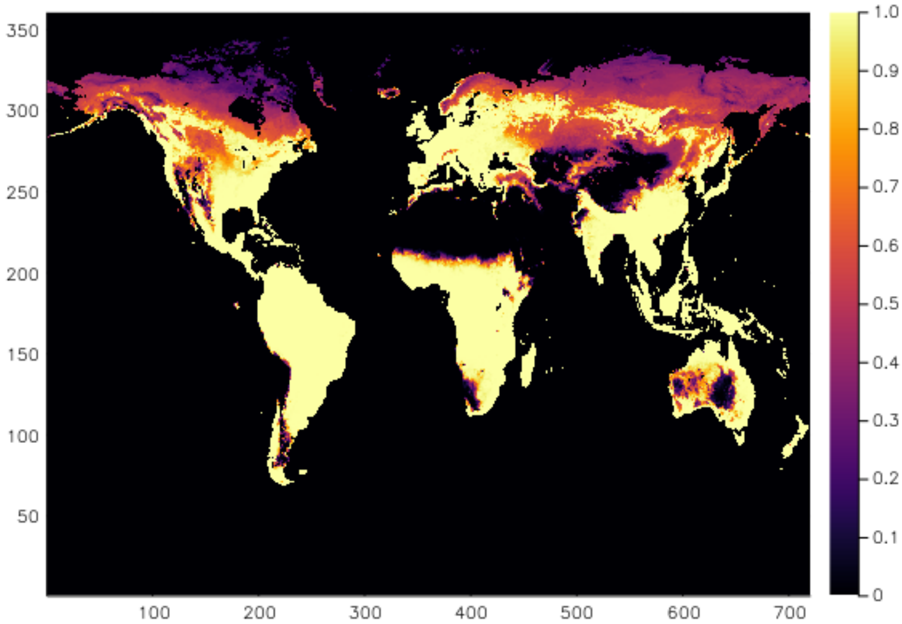


Fig. 1 - Variance per time scale

In [27]:
```
import ESDL.NaNMissing
indims = InDims(TimeAxis, miss=NaNMissing())
outdims = OutDims()
```

Out[27]:
```
ESDL.DAT.OutDims((), (), ESDL.DAT.DataArrayMissing(), zero, identity, :auto, f
alse, ESDL.DAT.AsArray(), 1)
```

In [28]:
```
# calculate variance for a cube taking into account a mask (here for NDVI)

function cubevar_mask(xout, xin, mask; nmin=10)
    # find indices with valid values, which are not masked
    xidx = .!isnan.(xin) .& .!isnan.(mask)
    # return NaN if less than nmin value pairs are valid, else return variance
    xout[:] = count(xidx) < nmin ? NaN : var(xin[xidx])
    return xout[:]
end
```

Out[28]:
```
cubevar_mask (generic function with 1 method)
```

In [29]:
```julia
# calculate variance for a cube without need for masking (Prec, Tair)

function cubevar(xout, xin; nmin=10)
    # find indices with valid values, which are not masked
    xidx = .!isnan.(xin)
    # return NaN if less than nmin value pairs are valid, else return variance
    xout[:] = count(xidx) < nmin ? NaN : var(xin[xidx])
    return xout[:]
end
```

Out[29]:
```
cubevar (generic function with 1 method)
```

In [30]:
```julia
# variance of NDVI using a mask of valid values

c_fft_var_vegidx = mapCube(cubevar_mask, (c_fft_vegidx,cubemask_gapfilled), in
dims=(indims,indims), outdims=outdims)
```

```
Progress: 100%|███████████████████████████████████████| Time: 0:01:00:29
```

Out[30]:
```
In-Memory data cube with the following dimensions
Scale                Axis with elements: Trend Long-Term Variability Annual Cyc
le Fast Oscillations
Lon                  Axis with 720 Elements from -179.75 to 179.75
Lat                  Axis with 360 Elements from 89.75 to -89.75
Total size: 8.9 MB
```

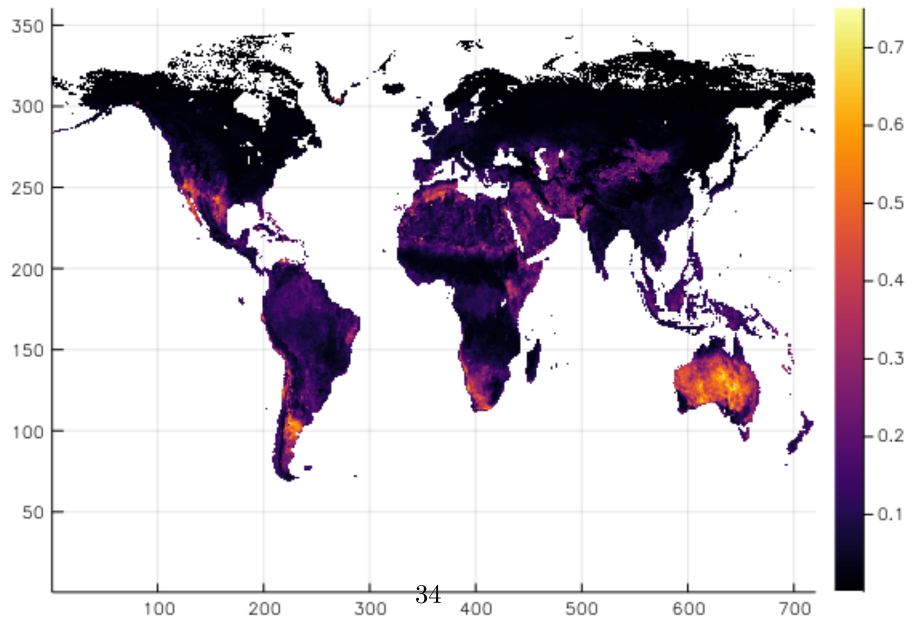
In [31]:
```julia
c_fft_var_prec = mapCube(cubevar, c_fft_prec, indims=indims, outdims=outdims)
c_fft_var_temp = mapCube(cubevar, c_fft_temp, indims=indims, outdims=outdims)
```

```
Progress: 100%|███████████████████████████████████████| Time: 0:00:45
Progress: 100%|███████████████████████████████████████| Time: 0:00:45
```

Out[31]:
```
In-Memory data cube with the following dimensions
Scale                Axis with elements: Trend Long-Term Variability Annual Cyc
le Fast Oscillations
Lon                  Axis with 720 Elements from -179.75 to 179.75
Lat                  Axis with 360 Elements from 89.75 to -89.75
Total size: 8.9 MB
```

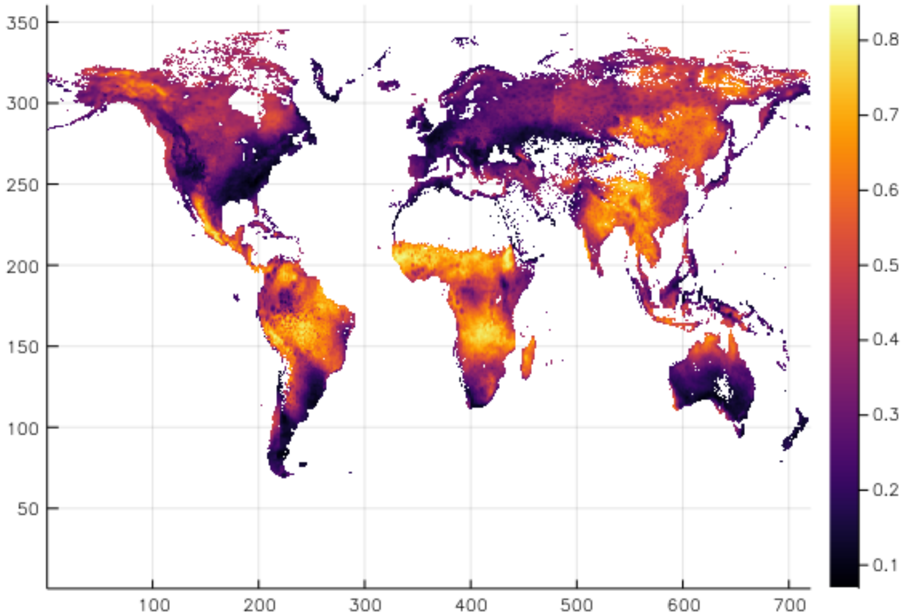
In [32]:
```julia
# Example maps:
# NDVI longer-term variability map
heatmap(c_fft_var_vegidx[2,:,:]'[end:-1:1,:])
```

Out[32]:



In [33]:
```
# precipication seasonal(annual) variability map masked by ocean and desert ma
sk
heatmap((c_fft_var_prec[3,:,:].*oce_des_mask)'[end:-1:1,:])
```

Out[33]:



In [34]:
```
# temperature short-term variability map
heatmap((c_fft_var_temp[4,:,:].*oce_des_mask)'[end:-1:1,:])
```

Out[34]:

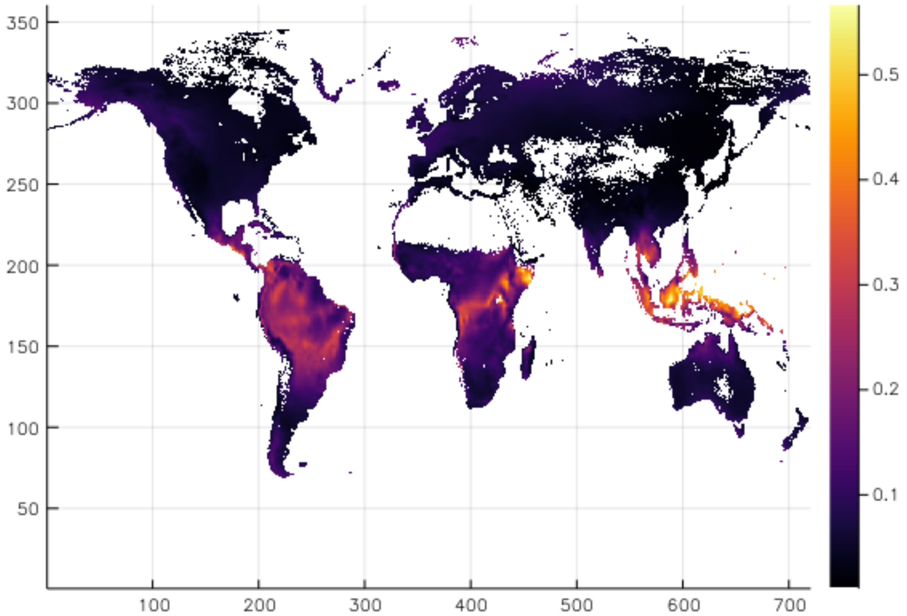


Fig. 2 - Dominant Scale of Variation

In [35]:
```
# Find TS of maximum variance, ignoring NaN

function findmax_nan(x)
    if all(isnan.(x))
        return NaN
    else
        return findmax(x)[2]
    end
end
```

Out[35]:
```
findmax_nan (generic function with 1 method)
```

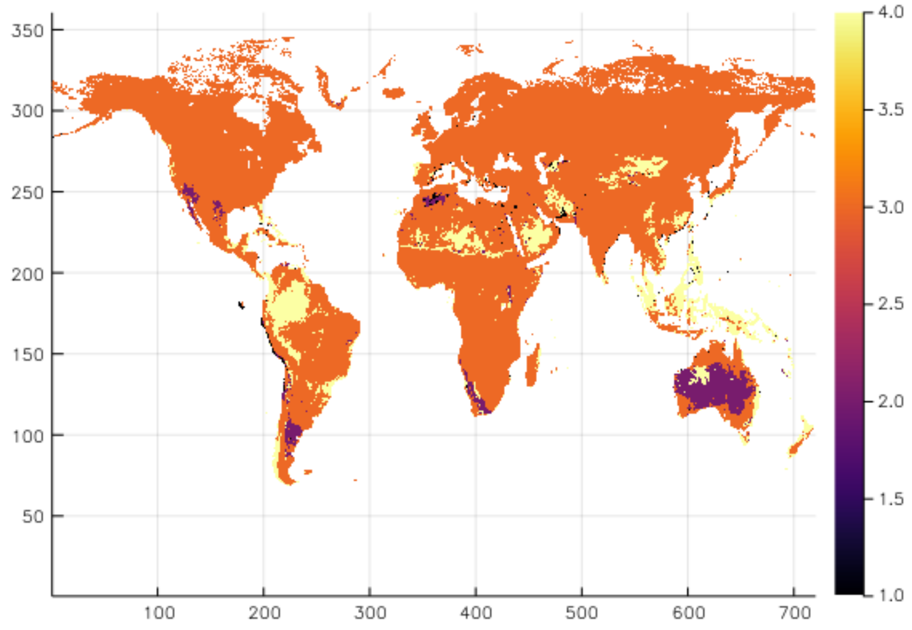
In [36]:
```
c_fft_max_vegidx = mapslices(x->findmax_nan(x), c_fft_var_vegidx, "Scale")
c_fft_max_prec = mapslices(x->findmax_nan(x), c_fft_var_prec, "Scale")
c_fft_max_temp = mapslices(x->findmax_nan(x), c_fft_var_temp, "Scale")
```

```
Progress: 100%|█████████████████████████████████████████| Time: 0:00:05
Progress: 100%|█████████████████████████████████████████| Time: 0:00:05
Progress: 100%|█████████████████████████████████████████| Time: 0:00:05
```

Out[36]:
```
In-Memory data cube with the following dimensions
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 2.22 MB
```

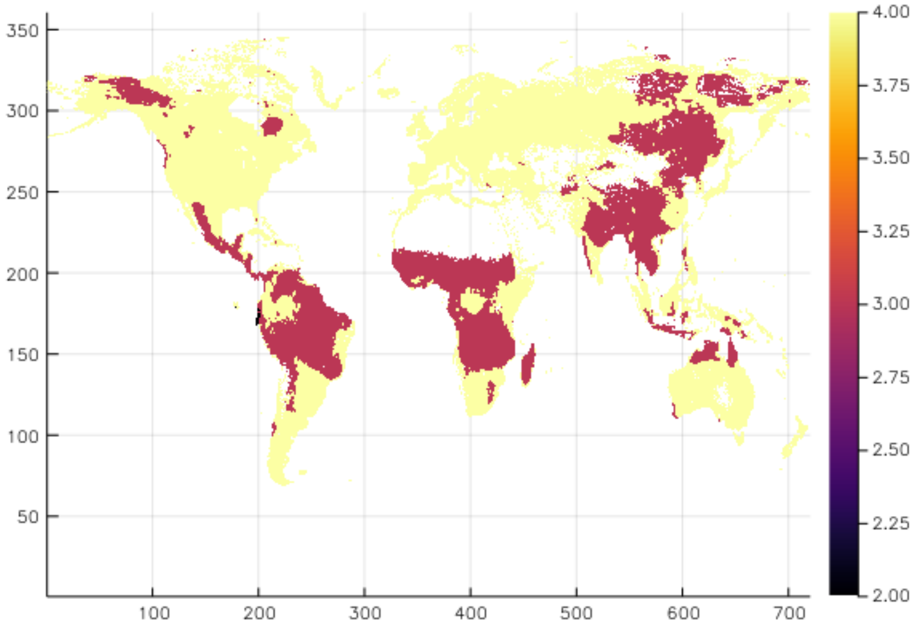
In [37]:
```
# map dominant scale of variability in NDVI
# 1 = Trend; 2 = Longer-term, 3 = Annual cycle, 4 = Short-term

heatmap(c_fft_max_vegidx[:,:]'[end:-1:1,:])
```

Out[37]:



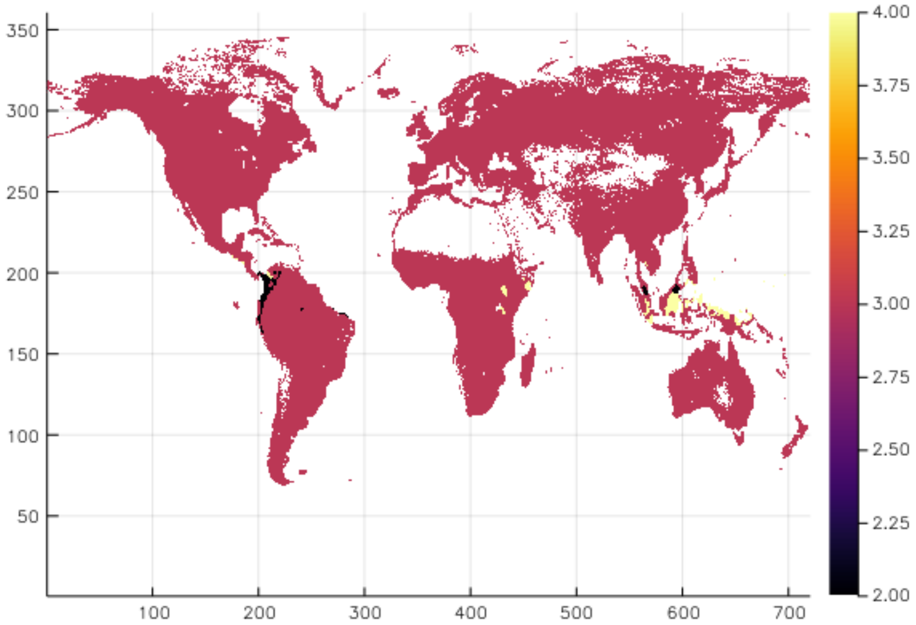
In [38]:
```
# map dominant scale of variability in precipitation
# 1 = Trend; 2 = Longer-term, 3 = Annual cycle, 4 = Short-term

heatmap((c_fft_max_prec[:,:].*oce_des_mask)'[end:-1:1,:])
```

Out[38]:



In [39]:
```
# map dominant scale of variability in temperature
# 1 = Trend; 2 = Longer-term, 3 = Annual cycle, 4 = Short-term

heatmap((c_fft_max_temp[:,:].*oce_des_mask)'[end:-1:1,:])
```

Out[39]:



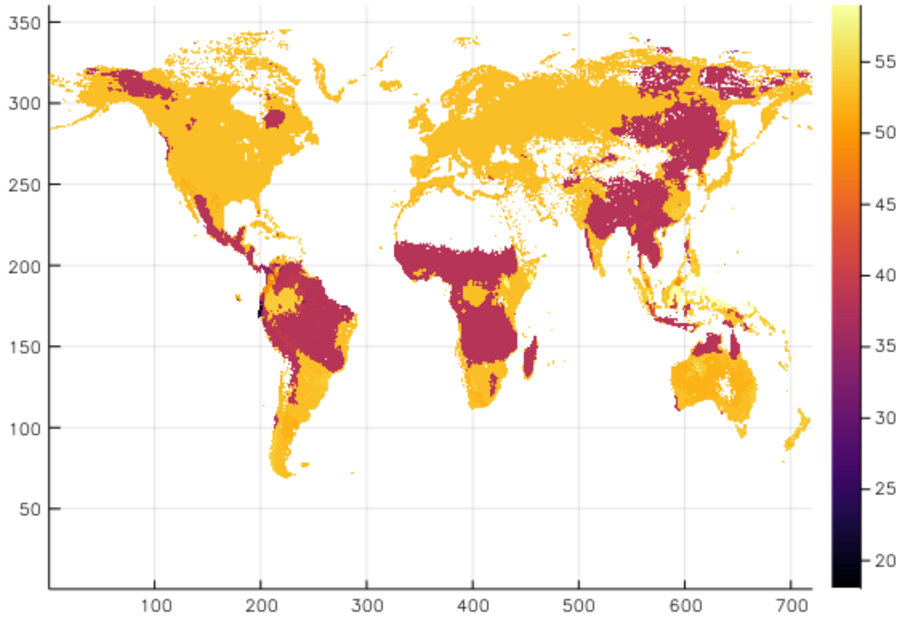
Create combinations of classes

In [40]:
```
x1 = c_fft_max_vegidx.data
x2 = c_fft_max_temp.data
x3 = c_fft_max_prec.data;
```

In [41]:
```
using RCall

# combine dominant scales

# get numeric variable for dominant scales
x123 = rcopy(R"as.numeric(interaction($x1, $x2, $x3))")
xclasses_numeric = reshape(x123, size(x1,1), size(x1,2));
```

In [43]:
```
# map combination of dominant scales of variability
heatmap((xclasses_numeric[:,:].*oce_des_mask)'[end:-1:1,:])
```

Out[43]:



In [44]:
```
# translate numeric to oscialltion class
x_comb = unique(rcopy(R"interaction($x1, $x2, $x3)"))

x_comb_leg = x_comb |>
(y->replace.(y,"1" => "T")) |>
(y->replace.(y,"2" => "L")) |>
(y->replace.(y,"3" => "A")) |>
(y->replace.(y,"4" => "S"));
```

In [45]:
```
domscale_leg = Dict(zip(unique(rcopy(R"as.numeric(interaction($x1, $x2, $x
3))")),x_comb_leg))

# delete unwanted categories containing NaN
get.(domscale_leg,collect(0:5:60),0)
delete!.(domscale_leg,collect(0:5:60))

domscale_leg
```

Out[45]:
```
Dict{Float64,String} with 26 entries:
  18.0 => "A.L.L"
  54.0 => "S.A.S"
  39.0 => "S.A.A"
  21.0 => "T.A.L"
  43.0 => "A.S.A"
  58.0 => "A.S.S"
  34.0 => "S.L.A"
  59.0 => "S.S.S"
  8.0  => "A.A.T"
  51.0 => "T.A.S"
  6.0  => "T.A.T"
  44.0 => "S.S.A"
  49.0 => "S.L.S"
  37.0 => "L.A.A"
  24.0 => "S.A.L"
  22.0 => "L.A.L"
  53.0 => "A.A.S"
  38.0 => "A.A.A"
  57.0 => "L.S.S"
  23.0 => "A.A.L"
  31.0 => "T.L.A"
  36.0 => "T.A.A"
  48.0 => "A.L.S"
  56.0 => "T.S.S"
  52.0 => "L.A.S"
  ⋮    => ⋮
```

In [46]:
```
#using ESDL

c_dom_scale = ESDL.CubeMem(
    CubeAxis[c_vegidx.axes[1],c_vegidx.axes[2]],
    xclasses_numeric,
    # mask categories containing NaN
    map(i->i in (10,20,25,30,35,40,45,50,55,60) ? ESDL.Mask.MISSING : ESDL.Mas
k.VALID,xclasses_numeric))
```

Out[46]:
```
In-Memory data cube with the following dimensions
Lon                 Axis with 720 Elements from -179.75 to 179.75
Lat                 Axis with 360 Elements from 89.75 to -89.75
Total size: 2.22 MB
```

In [47]:
```julia
# map dominant co-oscillation regimes

heatmap(c_dom_scale[:,:]'[end:-1:1,:])
```

Out[47]:

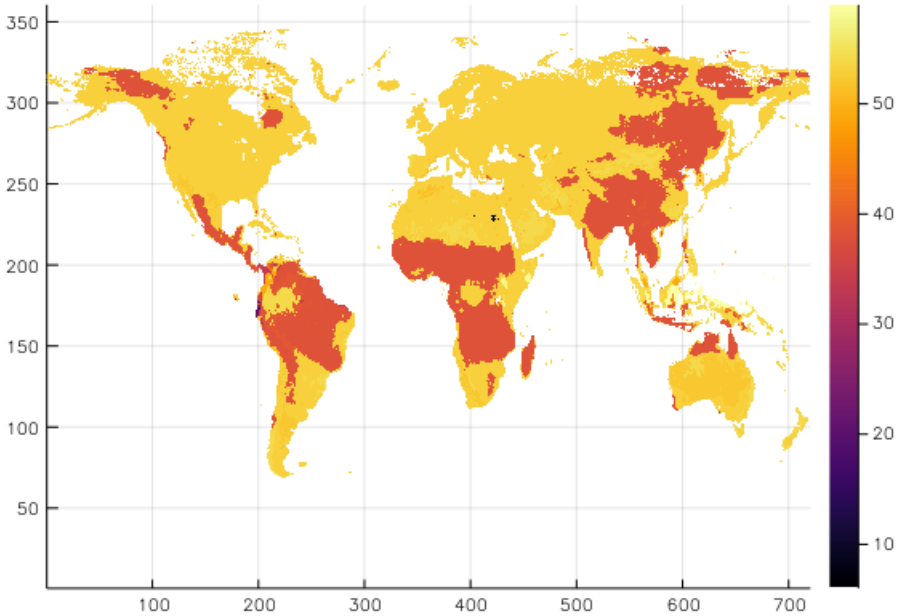


Fig. 3 - Correlations between variables

In [48]:
```julia
import ESDL.NaNMissing
indims = InDims("Time", miss=NaNMissing()), InDims("Time", miss=NaNMissing()),
InDims("Time", miss=NaNMissing())
outdims = OutDims()
```

Out[48]:
```
ESDL.DAT.OutDims((), (), ESDL.DAT.DataArrayMissing(), zero, identity, :auto, f
alse, ESDL.DAT.AsArray(), 1)
```

In [49]:
```julia
function cubecor_mask(xout,xin1,xin2,mask;nmin=10)
    # find indices with valid values, which are not masked
    xidx = .!isnan.(xin1) .& .!isnan.(xin2) .& .!isnan.(mask)
    # return NaN if less than nmin value pairs are valid, else return correlat
ion
    xout[:] = count(xidx) < nmin ? NaN : cor(xin1[xidx],xin2[xidx])
    return xout[:]
end
```

Out[49]:
```
cubecor_mask (generic function with 1 method)
```

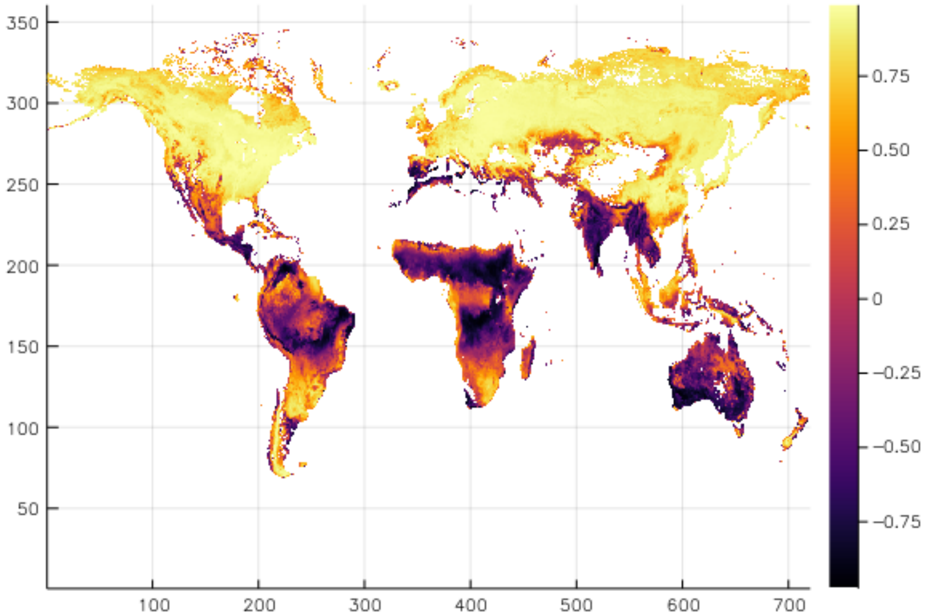
In [50]:
```julia
cor_vegidx_temp = mapCube(cubecor_mask, (c_fft_vegidx, c_fft_temp,cubemask), i
ndims = indims, outdims = outdims)
```

```
Progress: 100%|██████████████████████████████████████████| Time: 0:01:268:16
```

Out[50]:
```
In-Memory data cube with the following dimensions
Scale                Axis with elements: Trend Long-Term Variability Annual Cyc
le Fast Oscillations
Lon                  Axis with 720 Elements from -179.75 to 179.75
Lat                  Axis with 360 Elements from 89.75 to -89.75
Total size: 8.9 MB
```

In [51]:
```
# correlation between NDVI and air temperature at the annual scale

heatmap(cor_vegidx_temp[3,:,:]'[end:-1:1,:])
```

Out[51]:



In [52]:
```
function cubecor_lagged_mask(xout,xin1,xin2,mask;nmin=10,lag=1)
    x = xin1[1+lag:end] # lag first time series (and corresponding mask)
    m = mask[1+lag:end]
    y = xin2[1:end-lag] # cut second time series by lag
    xidx = .!isnan.(x) .& .!isnan.(y) .& .!isnan.(m) # indices with valid valu
e pairs, not masked
    # return NaN if less than nmin value pairs are valid, else return correlat
ion
    xout[:] = count(xidx) < nmin ? NaN : cor(x[xidx],y[xidx])
    return xout[:]
end
```

Out[52]:
```
cubecor_lagged_mask (generic function with 1 method)
```

In [53]:
```
cor_vegidx_prec_lag1 = mapCube(cubecor_lagged_mask,(c_fft_vegidx,c_fft_prec,cu
bemask),
    indims = indims, outdims = outdims)
```

```
Progress: 100%|██████████████████████████████████████████| Time: 0:01:35
```

Out[53]:
```
In-Memory data cube with the following dimensions
Scale                Axis with elements: Trend Long-Term Variability Annual Cyc
le Fast Oscillations
Lon                  Axis with 720 Elements from -179.75 to 179.75
Lat                  Axis with 360 Elements from 89.75 to -89.75
Total size: 8.9 MB
```

In [54]:
```
# correlation between NDVI and precipitation at the annual oscillation (lagged
1 time-step)
heatmap(cor_vegidx_prec_lag1[3,:,:]'[end:-1:1,:])
```

Out[54]:

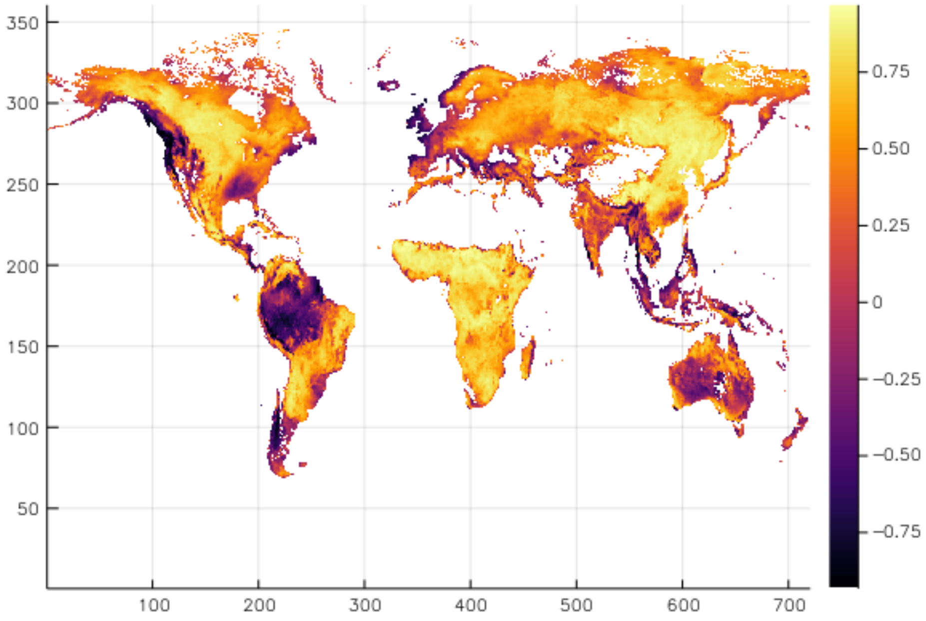


Fig. 3 - Bicolor Maps

In [55]:
```
# function to separate data within different quantiles

function quaId2(arr, qtls)
    y = zeros(size(arr))./0      # NaN output array of size input array
    for i=(size(qtls)[1]):-1:2   # each quantile boundary, counting downwards
        y[arr .< qtls[i]]  = i.-1   # wherever x-value .< quantile[i] is true,
set y to i-1
                # (below boundary 2 == 1st quantile)
    end
    return y
end
```

Out[55]: quaId2 (generic function with 1 method)

In [56]:
```
## quantile boundaries
qua=[-1,-0.6,-0.2,0.2,0.6,1]
```

Out[56]:
```
6-element Array{Float64,1}:
 -1.0
 -0.6
 -0.2
  0.2
  0.6
  1.0
```

In [57]:
```julia
## Generate different quatiles ID for all variables and each time scale

precTrend = quaId2(cor_vegidx_prec_lag1[1,:,:],qua)
precLongTerm = quaId2(cor_vegidx_prec_lag1[2,:,:],qua)
precAnnual = quaId2(cor_vegidx_prec_lag1[3,:,:],qua)
precFastOs = quaId2(cor_vegidx_prec_lag1[4,:,:],qua)

tempTrend = quaId2(cor_vegidx_temp[1,:,:],qua)
tempLongTerm = quaId2(cor_vegidx_temp[2,:,:],qua)
tempAnnual = quaId2(cor_vegidx_temp[3,:,:],qua)
tempFastOs = quaId2(cor_vegidx_temp[4,:,:],qua);
```

In [58]:
```julia
# based on the X and Y variables the color ID is defined, later translated int
o color for plotting
colId = reshape(1:25,5,5)'
```

Out[58]:
```
5×5 Array{Int64,2}:
  1   2   3   4   5
  6   7   8   9  10
 11  12  13  14  15
 16  17  18  19  20
 21  22  23  24  25
```

In [59]:
```julia
## This function establishes the respective combinations between
## quantiles and bicolor scale for two input maps

function BicolorMap(A,B)
    xout = fill(NaN, size(A)) # empty output array
    for x in 1:(size(A)[1]) # looping through latitudes
        for y in 1:(size(A)[2])  # looping through longitudes
            if !isnan(A[x,y]) && !isnan(B[x,y])
            i = Integer(A[x,y])
            j = Integer(B[x,y])
            colIdPx = colId[i,j]
            xout[x,y] = colIdPx
            end
       end
    end
    return(xout)
end
```

Out[59]:
```
BicolorMap (generic function with 1 method)
```

In [60]:
```julia
trendCor=BicolorMap(tempTrend, precTrend)
longTermCor=BicolorMap(tempLongTerm, precLongTerm)
annualCor=BicolorMap(tempAnnual, precAnnual)
fastOsCor=BicolorMap(tempFastOs, precFastOs);
```

In [61]:
```
# "bicolor-map" of longer-term correlation of temperature
# and precipitation

heatmap(longTermCor[:,:]'[end:-1:1,:])
```

Out[61]:

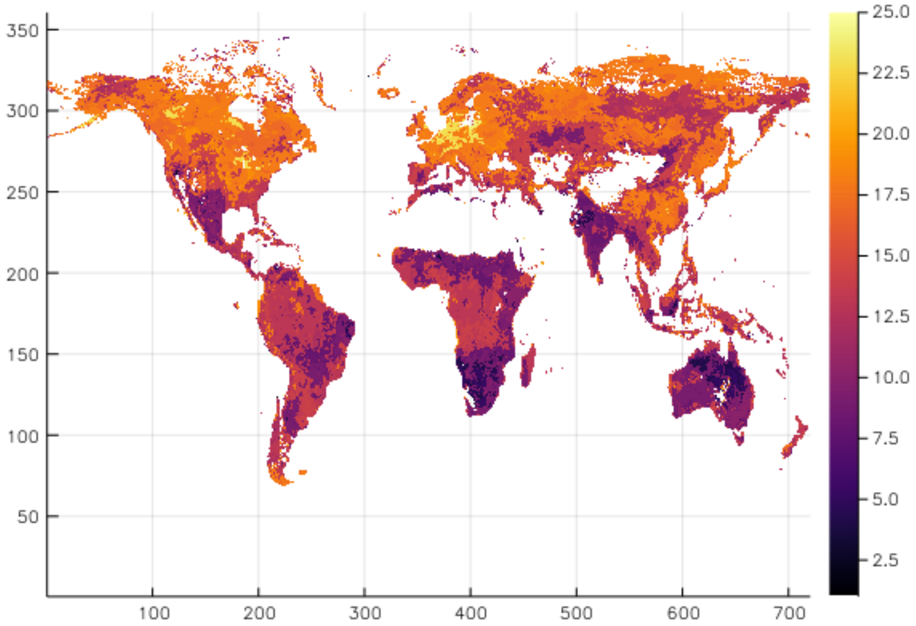
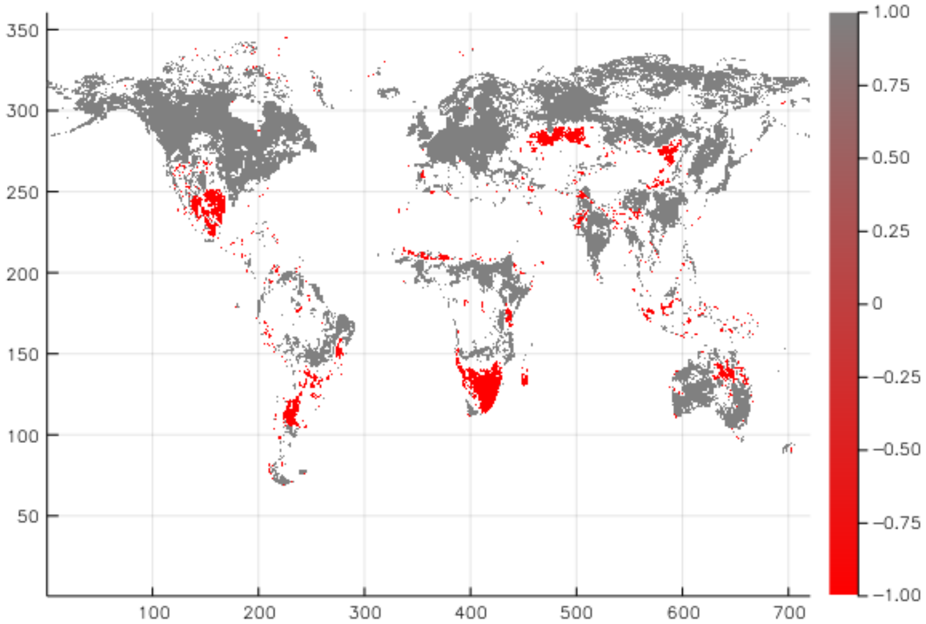


Fig. 4 - Comparison of differences in the correlation sign

In [62]:
```
# reclassify values into a binary map, 1 = positive correlations and -1 = nega
tive correlations
# correlations between -0.2 and 0.2 are ommited (NaN)

cor_vegidx_temp_sign = map(x-> x > 0.2 ? 1 : x < -0.2 ? -1 : NaN, cor_vegidx_t
emp[:,:,:]);
cor_vegidx_prec_sign = map(x-> x > 0.2 ? 1 : x < -0.2 ? -1 : NaN, cor_vegidx_p
rec_lag1[:,:,:]);
```

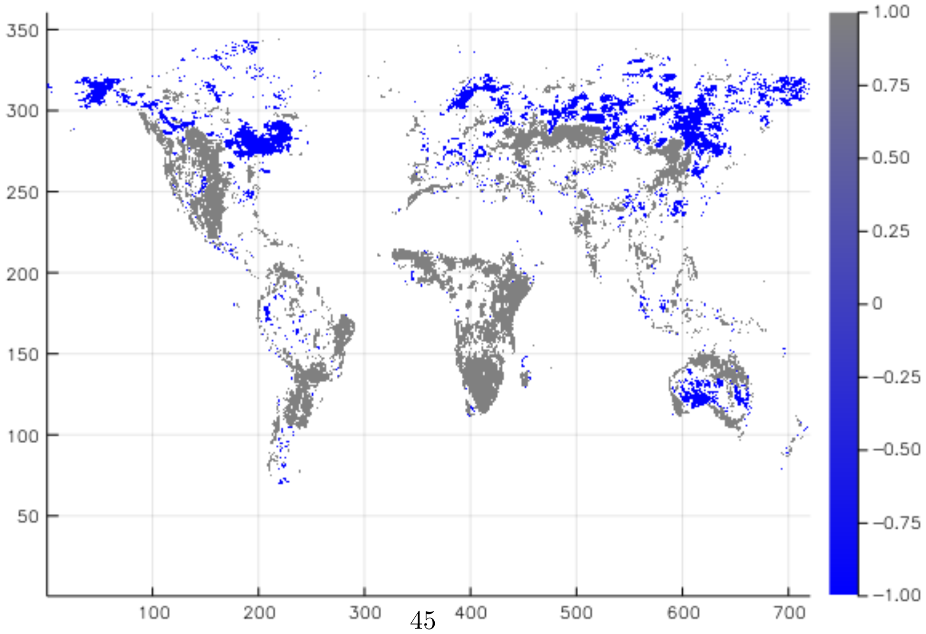
In [63]:
```
# comparison of differences in the sign of correlation between annual and long
-term scales
# for air-temperature. 1 = equal correlation sign. -1 = different correlation
sign

cor_vegidx_temp_sign_ch = (cor_vegidx_temp_sign[2,:,:].*cor_vegidx_temp_sign
[3,:,:].*oce_des_mask)
heatmap(cor_vegidx_temp_sign_ch'[end:-1:1,:], color=cgrad([:red, :grey]))
```

Out[63]:



In [64]:
```
# comparison of differences in the sign of correlation between annual and long
-term scales
# for precipiatation. 1 = equal correlation sign. -1 = different correlation s
ign

cor_vegidx_prec_sign_ch = cor_vegidx_prec_sign[2,:,:].*cor_vegidx_prec_sign
[3,:,:].*oce_des_mask
heatmap(cor_vegidx_prec_sign_ch'[end:-1:1,:], color=cgrad([:blue, :grey]))
```

Out[64]:



In [65]:
```
# reclassify values of air temperature and preciptation for global comparison

# 2 = pixels with different correlation sign, otherwise 0
cor_vegidx_temp_ch_reclass = map(x-> x == -1.0 ? 2.0 : x == 1 ? 0.0 : NaN, cor
_vegidx_temp_sign_ch[:,:]);

# -1 = pixel with different correlation sign, otherwise 0
cor_vegidx_prec_ch_reclass = map(x-> x == -1.0 ? -1.0 : x == 1 ? 0.0 : NaN, co
r_vegidx_prec_sign_ch[:,:]);
```

In [66]:
```
# map for differences in the correlation sign between annual and long-term sca
le for NDVI and air
# temperature (red), precipitation (blue), and both (purple).

heatmap((cor_vegidx_prec_ch_reclass[:,:] .+ cor_vegidx_temp_ch_reclass
[:,:])'[end:-1:1,:], color=cgrad([:blue, :white, :purple, :red]))
```

Out[66]: