NBR_postMPT_with_preMPT_dataquality

gridcode for this notebook: `NBR_postMPT_with_preMPT_dataquality` (post-MPT her er tilbake til 700 kyrs BP)

Predictive asymmetry analysis for Late Pleistocene Earth system dynamics,

using lower quality data for comparability with pre-MPT results.

Written by Maria Salem, in supplement to master thesis.

In this notebook we compute the normalized predictive asymmetry in the post-MPT, using the same records as available for the post-MPT (1570-1250 ka BP), but with lower quality data, so that we can compare the results for the two periods. The aim is to see if we can detect a significant change in the coupling of the climate system between the two periods, or if the difference in results fall within the natural variability of the system.

We approach the problem with a null-hypothesis test: Our null-hypothesis is that there is no (information in the data of) change in causal coupling of the climate system components from the pre-MPT to the post-MPT. The null-hypothesis test design is to compute one iteration of predictive asymmetry for the full time series length on the `postMPT_1500` grid (1574.0 ka BP - 1250 ka BP, e.g. a time series length N = 325 data points), and compare this to a family of predictive asymmetries of the post-MPT. The family of post-MPT $\mathcal{A}$ is computed with random sequences that are all of the pre-MPT time series length (325 data points, e.g, same data quality). From this family of $\mathcal{A}$ we can define a conficence interval for causal couplings for the post-MPT. The null-hypothesis of no detectable change in causal coupling of the climate system components from pre-MPT to post-MPT, is rejected if the pre-MPT-iteration falls outside the 95% confidence interval of the post-MPT results.

For this analysis, we use the records that cover both post-MPT and pre-MPT dynamics. (references used as record labels, in bold):

- **Ins**/La2004: Numerical solution for northern hemisphere insolation (top of atmosphere solar flux mean for summer solstice at $65^{\circ} N$), by *Laskar et al. (2004)*.
- **LR04**: $\delta^{18}O$ global reference stack - a principal component analysis of 57 $\delta^{18}O$ records, by *Lisiecki & Raymo (2005)*.
- **EldSL**/E: Temperature deconvolution of $\delta^{18}O$ signal in marine sediment core from the Chatnam Rise (Pacific Ocean), spanning the last 1.5 Myrs, by *Elderfield et al. (2012)*.
- **MedSL**/R: Relative sea level at Straight of Gibraltar, spanning the last 5.3 Myrs (Rohling et al., 2014).
- **MarFe**/MG: Marine sediment record of Fe accumulation in the Southern Ocean spanning the last 4 Ma, by *Martinez-García et al. (2011)*.

Outline for this notebook:

1. **pre-MPT**: Compute predictive asymmetry in one iteration of of full time series length on the `post-MPT_1500` grid
2. **post-MPT**: Compute a family of predictive asymmetries with the same time series length (324 data points), using the random sequences test. This will give us a confidence interval for causal couplings for the post-MPT.
3. **Null-hypothesis test**: Plot the pre-MPT iteration of predictive asymmetry on top of the post-MPT conficence interval. If the pre-MPT $\mathcal{A}$ iteration falls outside the post-MPT 95% confidence interval, we reject the null-hypothesis.

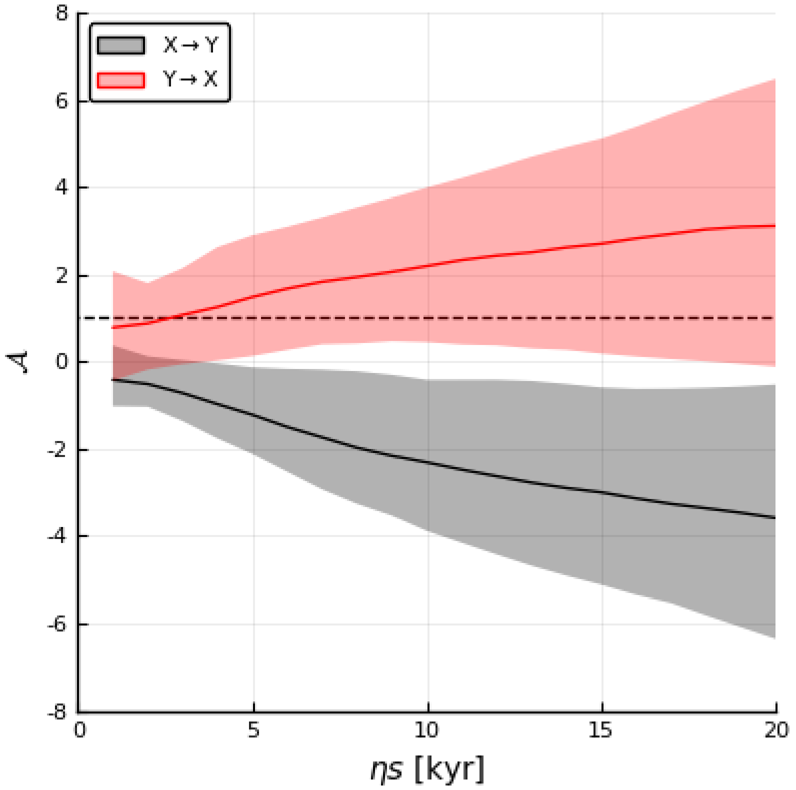
Important preamble: We import all necessary packages, wrangled time series and prededfined functions for analysis, by including notebook 3 `MA_NB3_Toolbox.ipynb`

In [1]:

```
using NBInclude
@nbinclude "../NBRs_ePalus_ns20/NB3_Toolbox_ns20.ipynb"
```

```
Bereiter BinnedResampled without interpolationThe LR04 d18O record s
pans from -5320.0 to 0.0 kyrs BP.
The La2004 insolation time series (Ins) spans from -5000.0 to -0.0 k
yrs BP.
Spratt Lisiecki  GSL stack (SpraSL) spans from -797.0 to -1.0 kyrs B
P.
The Elderfield GSL record (EldSL) spans from -1574.0 to -8.0 kyrs B
P.
The Grant sea level record (GraSL) spans from -491.0 to -1.0 kyrs B
P.
The Rohling sea level record (RohSL) spans from -5329.0 to -1.0 kyrs
BP.
The Bereiter pCO2 record (BerCO2) spans from -803.0 to -2.0 kyrs BP.
The Chalk pCO2 record (ChaCO2) spans from -1240.0 to -1092.0 kyrs B
P.
The Lambert dust record (IceDust) spans from -799.0 to -13.0 kyrs B
P.
The Martinez-Garcia Fe flux record (MarFe) spans from -3999.0 to -2.
0 kyrs BP.
The higher resolution part of the MG record spans from -800.0 to -0.
5 kyrs BP.The record is now on the common time gridThe record is now
on the common time gridResults are saved in the .jld2 file.  44.6637
71 seconds (165.45 M allocations: 11.168 GiB, 16.21% gc time)
```

Out[1]:



1. Cut the time series to the pre-MPT and post-MPT grids

1.1 Define the grids

In [2]:

```
#Recalling the same time intervals that are used for the other analyses

gridstart_preMPT = Int(maximum([tmin_E]))
gridend_preMPT   = Int(-1250) # Onset of the MPT

gridstart_postMPT = Int(maximum([tmin_G]))
gridend_postMPT   = Int(minimum(tmax_L))

#gridstart_postMPT = Int(-700)
#gridend_postMPT   = Int(minimum([tmax_LR04, tmax_E, tmax_MG, tmax_La2004]))

binsize_1 = 1   # 1 kyrs - is the default timestep on which all time series are
 binned (NB1)


# the common grids for the time series are then defined by
#### preMPT
commongrid_preMPT = gridstart_preMPT : binsize_1 : gridend_preMPT # bin midpoint
s defined as a grid
bmpArray_preMPT   = [commongrid_preMPT[i] for i in 1:length(commongrid_preMPT)]
# bin midpoints defined as an array
#### post-MPT
commongrid_postMPT = gridstart_postMPT : binsize_1 : gridend_postMPT # bin midpo
ints defined as a grid
bmpArray_postMPT    = [commongrid_postMPT[i] for i in 1:length(commongrid_postMP
T)] # bin midpoints defined as an array


print("Pre-MPT grid is defined by ", commongrid_preMPT , ".
Post-MPT grid is defined by ", commongrid_postMPT,"." )

#= *Note*:
the common grid for the time series in this notebook is defined by tmin and tmax
as the *bin midpoints*
(opposite to NB1, where tmin and tmax defined the *grid edges* for binned resamp
ling.) =#
```

```
Pre-MPT grid is defined by -1574:1:-1250.
Post-MPT grid is defined by -491:1:-13.
```

1.2 Cut the time series to the defined time intervals.

We select the time series' common time array as following: all time values greater (younger) than, or equal to, the common grid's starting midpoint `tmin` , and all values smaller (older) than, or equal to, the common grid's ending midpoint `tmax` .

In [3]:

```
#= After several attempts, we see that the >= (greater than OR EQUAL TO) operato
r does not seem to work for uivDs (our time series format, which we want to cu
t).
We will therefore set the the grid edges tmin and tmax a littlebit before and af
ter the start and end of the common grid, so that we don't lose the datapoints o
n the edges =#
tmin_preMPT = gridstart_preMPT - 0.01
tmax_preMPT = gridend_preMPT + 0.01

tmin_postMPT = gridstart_postMPT - 0.01
tmax_postMPT = gridend_postMPT + 0.01

#= select the time series' common time array as following:
 all values greater (younger) than ``tmin``, and smaller (older) than ``tmax`` =
#
```

Out[3]:

```
-12.99
```

Insolation - La2004

La2004 is a numerical solution for insolation, with no associated uncertainty for the time interval here observed. Therefore no confidence. interval on this record

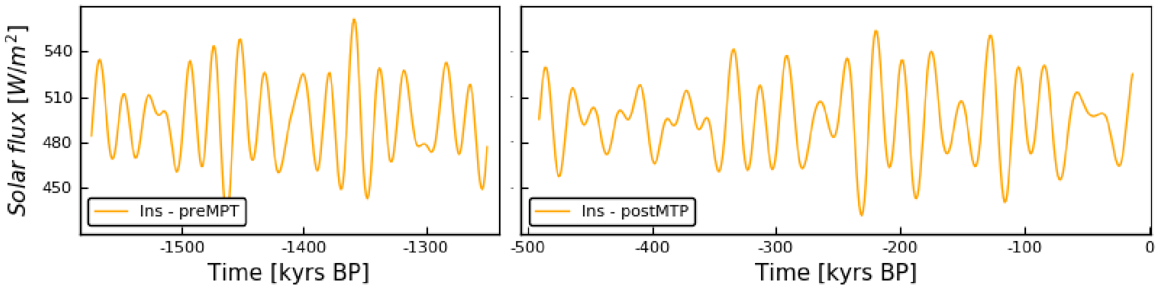
- Default version (`La2004_insol_cut`) is on a grid with 1 kyr timestep between each insolation value (as in the original dataset).

In [4]:

```
######### pre-MPT

# cut out the relevant time interval (tmin:tmax) from the time array
La2004_t_cut_preMPT = La2004_t_fullength[(La2004_t_fullength .>= gridstart_preMPT) .& (La2004_t_fullength .<= gridend_preMPT)]
# cut out the relevant time interval from the insolation values array
La2004_insol_cut_preMPT = La2004_insol65N_fullength[(La2004_t_fullength .>= gridstart_preMPT) .& (La2004_t_fullength .<= gridend_preMPT)]

# check that we have cut the correct time interval
print(La2004_t_cut_preMPT[1] : La2004_t_cut_preMPT[end] , " - the cut version of La2004 time series...
", gridstart_preMPT : gridend_preMPT , " - ...is now the same as the pre-MPT grid.") # so all good


plot_La2004_preMPT =
plot(La2004_t_cut_preMPT, La2004_insol_cut_preMPT,
    color = :orange,
    label = "Ins - preMPT",
    xlabel = "Time [kyrs BP]",  xticks = (-2000:100:0),
    ylabel = L"Solar \ flux \ [W/m^{2}]",
    legend = :topleft,
    grid = false,
    size = (500,200)
    )


########## post-MPT

# cut out the relevant time interval (tmin:tmax) from the time array
La2004_t_cut_postMPT = La2004_t_fullength[(La2004_t_fullength .>= gridstart_postMPT) .& (La2004_t_fullength .<= gridend_postMPT)]
# cut out the relevant time interval from the insolation values array
La2004_insol_cut_postMPT = La2004_insol65N_fullength[(La2004_t_fullength .>= gridstart_postMPT) .& (La2004_t_fullength .<= gridend_postMPT)]

# check that we have cut the correct time interval
print(La2004_t_cut_postMPT[1] : La2004_t_cut_postMPT[end] , " - the cut version of La2004 time series...
", gridstart_postMPT : gridend_postMPT , " - ...is now the same as the pre-MPT grid.") # so all good

plot_La2004_postMPT =
plot(La2004_t_cut_postMPT, La2004_insol_cut_postMPT,
    color = :orange, label = "Ins - postMTP",
    xlabel = "Time [kyrs BP]",  xticks = (-2000:100:0),
    ylabel = "", # L"Solar \ flux \ [W/m^{2}]",
    legend = :topleft, grid = false,
    size = (500,200) , ytickfont = false
    )

######### join the plots
l = @layout [a b{0.6w}]
p_La2004_preVSpost = plot(plot_La2004_preMPT, plot_La2004_postMPT, ylims = (420,570), layout = l, size = (800,200), legend = :bottomleft, border = true)
```

```
-1574.0:1.0:-1250.0 - the cut version of La2004 time series...
-1574:-1250 - ...is now the same as the pre-MPT grid.-491.0:1.0:-13.
0 - the cut version of La2004 time series...
-491:-13 - ...is now the same as the pre-MPT grid.
```

Out[4]:



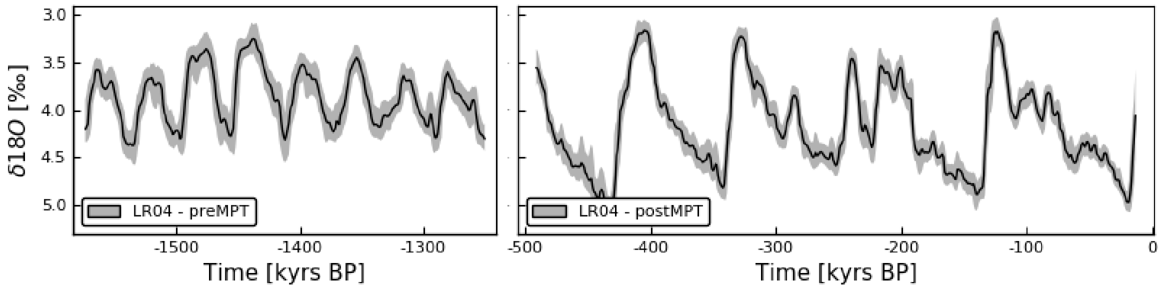
$\delta 18O$ - LR04

In [5]:

```
######### cut and plot the pre-MPT and post MPT intervals

# select the pre-MPT time window from the time series
LR04_cut_preMPT = cut_timeinterval_from_uivD(LR04_binned_fullength_fullageunc, Int(gridstart_preMPT), Int(gridend_preMPT), commongrid_preMPT)

# computing the median in each bin (0.5 quantile), and the confidence interval we want to use (95%)
bin_median = quantile.(LR04_cut_preMPT.values, 0.5)
bin_upperq = quantile.(LR04_cut_preMPT.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(LR04_cut_preMPT.values, 0.025)

#### Plot time series with the 95% confidence interval
plot_LR04_preMPT =
plot(bmpArray_preMPT, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3,
    color = :black, label =  "LR04 - preMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  L"\delta{18}O \ [\perthousand]",
    legend = :topleft,grid = false, size = (500,200), yflip = true)


####### post-MPT

# select the post-MPT time window from the time series
LR04_cut_postMPT = cut_timeinterval_from_uivD(LR04_binned_fullength_fullageunc, Int(gridstart_postMPT), Int(gridend_postMPT), commongrid_postMPT)

# computing the median in each bin (0.5 quantile), and the confidence interval we want to use (95%)
bin_median = quantile.(LR04_cut_postMPT.values, 0.5)
bin_upperq = quantile.(LR04_cut_postMPT.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(LR04_cut_postMPT.values, 0.025)

#### Plot time series with the 95% confidence interval
plot_LR04_postMPT =
plot(bmpArray_postMPT, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3,
    color = :black, label =  "LR04 - postMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  "", # L"\delta{18}O \ [\perthousand]",
    legend = :topleft, grid = false,
    size = (500,200), yflip = true,
    yaxis = :on, ytickfont = false
)

######### join the plots
l = @layout [a b{0.6w}]
p_LR04_preVSpost = plot(plot_LR04_preMPT, plot_LR04_postMPT, ylims = (2.9,5.3), layout = l, size = (800,200), legend = :bottomleft, border = true)
```

```
The record is now on the common time gridThe record is now on the co
mmon time grid
```

Out[5]:



Sea level - Elderfield record

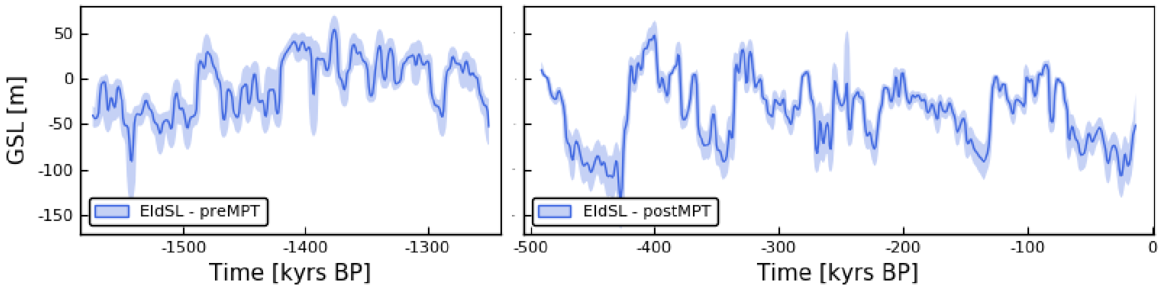
In [6]:

```
######## cut and plot the pre-MPT interval of the record
ts = E_binned_fullength_ageunc

# cut time series
ts_cut = cut_timeinterval_from_uivD(E_binned_fullength_ageunc, Int(gridstart_pre
MPT), Int(gridend_preMPT), commongrid_preMPT)
E_cut_preMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_E_preMPT =
plot(commongrid_preMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft,
    color = :royalblue, label =  "EldSL - preMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  "GSL [m]",
    grid = false, size = (500,200))


######### post-MPT

ts = E_binned_fullength_ageunc

# cut time series
ts_cut = cut_timeinterval_from_uivD(E_binned_fullength_ageunc, Int(gridstart_pos
tMPT), Int(gridend_postMPT), commongrid_postMPT)
E_cut_postMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_E_postMPT =
plot(commongrid_postMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft,
    color = :royalblue, label =  "EldSL - postMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  "", # "GSL [m]",
    grid = false, size = (500,200),
    ytickfont = false)

######### join the plots
l = @layout [a b{0.6w}]
```

```
p_E_preVSpost = plot(plot_E_preMPT, plot_E_postMPT, ylims = (-170,80),layout = l
, size = (800,200), legend = :bottomleft, border = true)
```

```
The record is now on the common time gridThe record is now on the co
mmon time grid
```

Out[6]:



dust - Martinez-García Fe MAR record

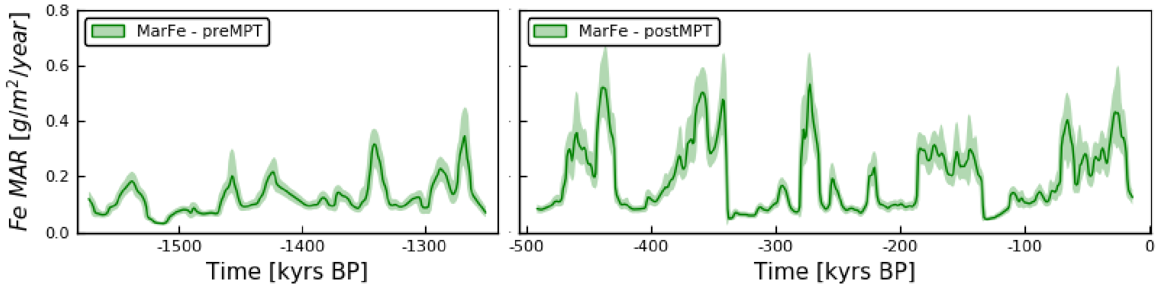
In [7]:

```
######## cut and plot the pre-MPT interval

ts = MG_binned_fullength

# cut time series
ts_cut = cut_timeinterval_from_uivD(ts, Int(gridstart_preMPT), Int(gridend_preMP
T), commongrid_preMPT)
MG_cut_preMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_MG_preMPT =
plot(commongrid_preMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :green,
    label =  "MarFe - preMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  L"Fe \ MAR \ [g/m^{2}/year]",
    grid = false
    )

########### post-MPT

ts = MG_binned_fullength

# cut time series
ts_cut = cut_timeinterval_from_uivD(ts, Int(gridstart_postMPT), Int(gridend_post
MPT), commongrid_postMPT)
MG_cut_postMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_MG_postMPT =
plot(commongrid_postMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :green,
    label =  "MarFe - postMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel = "", # L"Fe \ MAR \ [g/m^{2}/year]",
    grid = false, ytickfont = false, #ymirror = false, yaxis = :on
    )

######### join the plots
```

```
l = @layout [a b{0.6w}]
p_MG_preVSpost = plot(plot_MG_preMPT, plot_MG_postMPT, ylims = (0,0.8),layout =
l, size = (800,200), border = true)
```

```
The record is now on the common time gridThe record is now on the co
mmon time grid
```

Out[7]:



Make an overview plot of the time series

In [8]:

```
A4 = (8.27*72, 11.69*72) # A4 dimensions
A4h = (11.69*72, 8.27*72) # A4 dimensions,  horizontal layout

A3 = (8.27*72*3, 11.69*72*3) # A3 dimensions
```

Out[8]:

```
(1786.3199999999997, 2525.04)
```

In [9]:

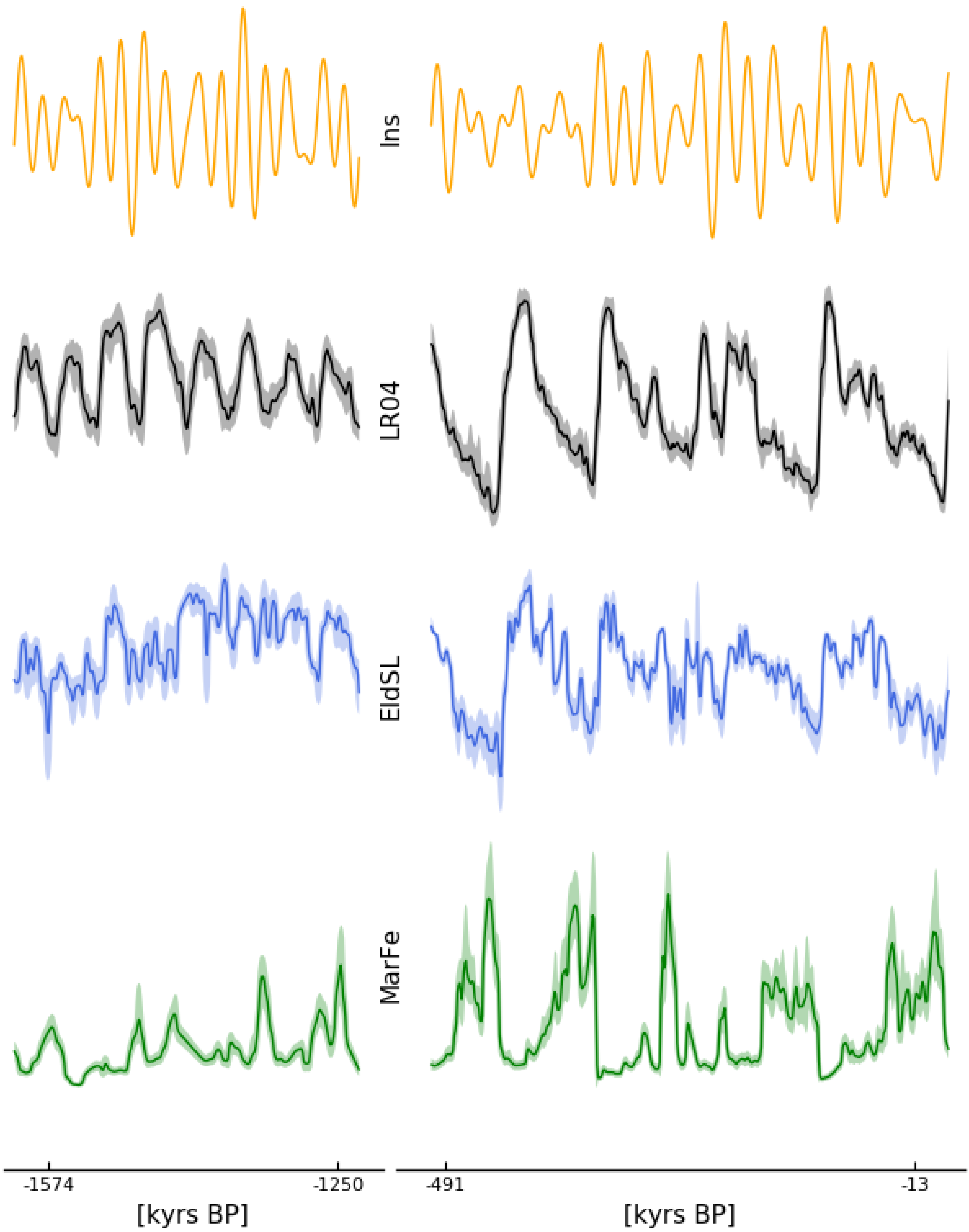
```
################################################ La2004
######### pre-MPT

# cut out the relevant time interval (tmin:tmax) from the time array
La2004_t_cut_preMPT = La2004_t_fullength[(La2004_t_fullength .>= gridstart_preMPT) .& (La2004_t_fullength .<= gridend_preMPT)]
# cut out the relevant time interval from the insolation values array
La2004_insol_cut_preMPT = La2004_insol65N_fullength[(La2004_t_fullength .>= gridstart_preMPT) .& (La2004_t_fullength .<= gridend_preMPT)]

# check that we have cut the correct time interval
print(La2004_t_cut_preMPT[1] : La2004_t_cut_preMPT[end] , " - the cut version of La2004 time series...
", gridstart_preMPT : gridend_preMPT , " - ...is now the same as the pre-MPT grid.") # so all good


plot_La2004_preMPT =
plot(title = "pre-MPT",
    La2004_t_cut_preMPT, La2004_insol_cut_preMPT,
    color = :orange,
    label = "Ins
preMPT",
    xlabel = "", #"Pre-MPT",
#Time [kyrs BP]",
    xmirror = true,
    xticks = (gridstart_preMPT:gridend_preMPT-gridstart_preMPT:gridend_preMPT),
#xticks = (-2000:100:0),
    ylabel = "Ins", #L"Solar \ flux \ [W/m^{2}]",
    legend = :topleft,
    grid = false,
    size = (500,200),
    ytickfont = false
    )


########## post-MPT

# cut out the relevant time interval (tmin:tmax) from the time array
La2004_t_cut_postMPT = La2004_t_fullength[(La2004_t_fullength .>= gridstart_postMPT) .& (La2004_t_fullength .<= gridend_postMPT)]
# cut out the relevant time interval from the insolation values array
La2004_insol_cut_postMPT = La2004_insol65N_fullength[(La2004_t_fullength .>= gridstart_postMPT) .& (La2004_t_fullength .<= gridend_postMPT)]

# check that we have cut the correct time interval
print(La2004_t_cut_postMPT[1] : La2004_t_cut_postMPT[end] , " - the cut version of La2004 time series...
", gridstart_postMPT : gridend_postMPT , " - ...is now the same as the pre-MPT grid.") # so all good

plot_La2004_postMPT =
plot(title = "post-MPT",
    La2004_t_cut_postMPT, La2004_insol_cut_postMPT,
    color = :orange, label = "Ins
postMTP",
    xlabel = "", #"Post-MPT",
#Time [kyrs BP]",
    xmirror = true,
    xticks = (gridstart_postMPT:gridend_postMPT-gridstart_postMPT:gridend_postMPT), #xticks = (-2000:100:0),
    ylabel = "", # L"Solar \ flux \ [W/m^{2}]",
    legend = :topleft,
    grid = false,
    size = (500,200),
    ytickfont = false
    )

######### join the La2004 plots
l = @layout [a b{0.6w}]
p_La2004_preVSpost = plot(plot_La2004_preMPT, plot_La2004_postMPT, ylims = (420, 570), layout = l, size = (800,200), legend = false, ytickfont = false, border = true,)



################################################## LR04

######### cut and plot the pre-MPT and post MPT intervals

# select the pre-MPT time window from the time series
LR04_cut_preMPT = cut_timeinterval_from_uivD(LR04_binned_fullength_fullageunc, Int(gridstart_preMPT), Int(gridend_preMPT), commongrid_preMPT)

# computing the median in each bin (0.5 quantile), and the confidence interval we want to use (95%)
bin_median = quantile.(LR04_cut_preMPT.values, 0.5)
bin_upperq = quantile.(LR04_cut_preMPT.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(LR04_cut_preMPT.values, 0.025)

#### Plot time series with the 95% confidence interval
plot_LR04_preMPT =
plot(bmpArray_preMPT, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3,
    color = :black, label =  "LR04
preMPT",
    xlabel =  "pre-MPT", #"Time [kyrs BP]",
    xticks = (gridstart_preMPT:gridend_preMPT-gridstart_preMPT:gridend_preMPT),
    ylabel = "LR04", #L"\delta{18}O \ [\perthousand]",
    legend = :topleft,grid = false, size = (500,200), yflip = true)


####### post-MPT

# select the post-MPT time window from the time series
LR04_cut_postMPT = cut_timeinterval_from_uivD(LR04_binned_fullength_fullageunc, Int(gridstart_postMPT), Int(gridend_postMPT), commongrid_postMPT)

# computing the median in each bin (0.5 quantile), and the confidence interval we want to use (95%)
bin_median = quantile.(LR04_cut_postMPT.values, 0.5)
bin_upperq = quantile.(LR04_cut_postMPT.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(LR04_cut_postMPT.values, 0.025)

#### Plot time series with the 95% confidence interval
plot_LR04_postMPT =
plot(bmpArray_postMPT, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3,
    color = :black, label =  "LR04
postMPT",
    xlabel =  "post-MPT", #"Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  "", # L"\delta{18}O \ [\perthousand]",
    xticks = (gridstart_postMPT:gridend_postMPT-gridstart_postMPT:gridend_postMP
T),
    legend = :topleft, grid = false,
    size = (500,200), yflip = true,
    yaxis = :on, ytickfont = false
)

######### join the LR04 plots
l = @layout [a b{0.6w}]
p_LR04_preVSpost = plot(plot_LR04_preMPT, plot_LR04_postMPT, ylims = (2.9,5.3),
layout = l, size = (800,200), legend = false, border = true, tickfont = false, x
label = "" )


################################################## EldSL

######## cut and plot the pre-MPT interval of the record
ts = E_binned_fullength_ageunc

# cut time series
ts_cut = cut_timeinterval_from_uivD(E_binned_fullength_ageunc, Int(gridstart_pre
MPT), Int(gridend_preMPT), commongrid_preMPT)
E_cut_preMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_E_preMPT =
plot(commongrid_preMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft,
    color = :royalblue, label =  "EldSL
preMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  "EldSL", #"GSL [m]",
    grid = false, size = (500,200))


######### post-MPT

ts = E_binned_fullength_ageunc

# cut time series
ts_cut = cut_timeinterval_from_uivD(E_binned_fullength_ageunc, Int(gridstart_pos
tMPT), Int(gridend_postMPT), commongrid_postMPT)
E_cut_postMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval
```

```
# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_E_postMPT =
plot(commongrid_postMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft,
    color = :royalblue, label =  "EldSL
postMPT",
    xlabel =  "Time [kyrs BP]", xticks = (-2000:100:0),
    ylabel =  "", # "GSL [m]",
    grid = false, size = (500,200),
    ytickfont = false)

######### join the EldSL plots
l = @layout [a b{0.6w}]
p_E_preVSpost = plot(plot_E_preMPT, plot_E_postMPT, ylims = (-170,80),layout = l
, size = (800,200), legend = false, border = true, xlabel = "", tickfont = false
)



############################################################### MarFe

######## cut and plot the pre-MPT interval

ts = MG_binned_fullength

# cut time series
ts_cut = cut_timeinterval_from_uivD(ts, Int(gridstart_preMPT), Int(gridend_preMP
T), commongrid_preMPT)
MG_cut_preMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_MG_preMPT =
plot(commongrid_preMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :green,
    label =  "MarFe
preMPT",
    xlabel =  "[kyrs BP]",
    xticks = (gridstart_preMPT : gridend_preMPT - gridstart_preMPT : gridend_pre
MPT),
    ylabel =  "MarFe", #L"Fe \ MAR \ [g/m^{2}/year]",
    grid = false,
    ytickfont = false
    )
```

```
########### post-MPT

ts = MG_binned_fullength

# cut time series
ts_cut = cut_timeinterval_from_uivD(ts, Int(gridstart_postMPT), Int(gridend_postMPT), commongrid_postMPT)
MG_cut_postMPT = ts_cut # save cut time series in an unambiguous name


### Plot time series with the 95% confidence interval

# computing the median in each bin (0.5 quantile), and the confidence interval we want to use (95%)
bin_median = quantile.(ts_cut.values, 0.5)
bin_upper = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts_cut.values, 0.025)
;

plot_MG_postMPT =
plot(commongrid_postMPT, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :green,
    label =  "MarFe
postMPT",
    xlabel =  "[kyrs BP]",
    xticks = (gridstart_postMPT : gridend_postMPT - gridstart_postMPT : gridend_postMPT),
    ylabel = "", # L"Fe \ MAR \ [g/m^{2}/year]",
    grid = false,
    ytickfont = false, #ymirror = false, yaxis = :on
    )

######### join the plots
l = @layout [a b{0.6w}]
p_MG_preVSpost = plot(plot_MG_preMPT, plot_MG_postMPT, ylims = (0,0.7),layout = l, size = (800,200), border = true,)

#########

####################################################################################
####################################################################################
####################################################################################


p1 = plot(p_La2004_preVSpost, title= "",  xlabel = "", xtickfont = false,     xaxis = :off, yaxis = :off)
p2 = plot(p_LR04_preVSpost, xlabel = "", xtickfont = false,     xaxis = :off, yaxis = :off)
p3 = plot(p_E_preVSpost, xlabel = "", xtickfont = false,     xaxis = :off, yaxis = :off)
p4 = plot(p_MG_preVSpost,  xlabel = "", xtickfont = false,     xaxis = :off, yaxis = :off)

##
p_xaxis_preMPT =
plot(#title = "pre-MPT",
    xlabel =  "[kyrs BP]",  xlims = (gridstart_preMPT-50, gridend_preMPT+50),
    xticks = (gridstart_preMPT : gridend_preMPT - gridstart_preMPT : gridend_pre
MPT),
    ylims = (0,0), ylabel = "", yaxis = :off, grid = :off)
p_title_preMPT = plot(title = "pre-MPT",ylims = (0,0), xlabel = "", ylabel = "",
xaxis = :off, yaxis = :off, grid = :off)

p_xaxis_postMPT =
plot(#title = "post-MPT",
    xlabel =  "[kyrs BP]",  xlims = (gridstart_postMPT-50, gridend_postMPT+50),
    xticks = (gridstart_postMPT : gridend_postMPT - gridstart_postMPT : gridend_
postMPT),
    ylims = (0,0), ylabel = "",  yaxis = :off, grid = :off)
p_title_postMPT = plot(title = "post-MPT",ylims = (0,0), xlabel = "", ylabel =
"", xaxis=:off, yaxis = :off, grid = :off)


l = @layout[a b{0.6w}]
p_xaxis_preVSpost = plot(p_xaxis_preMPT, p_xaxis_postMPT, layout = l, xtickfont
= 8, xaxis = :true,)
p_title_preVSpost = plot(p_title_preMPT, p_title_postMPT, layout = l,)
##

l = @layout [a{0.05h}; b; c; d; e; f{0.05h}]
pe_timeseries =
plot(p_title_preVSpost,
    p1,p2,p3,p4,
    p_xaxis_preVSpost,
    layout = l,
    size = A4, #(400,600),
    #ymirror = false,
    #ylims = (-10,10),
    #yticks = (-20:5:20),
    legend = false,
    ymirror = true,
    )
```

```
-1574.0:1.0:-1250.0 - the cut version of La2004 time series...
-1574:-1250 - ...is now the same as the pre-MPT grid.-491.0:1.0:-13.
0 - the cut version of La2004 time series...
-491:-13 - ...is now the same as the pre-MPT grid.The record is now
on the common time gridThe record is now on the common time gridThe
record is now on the common time gridThe record is now on the common
time gridThe record is now on the common time gridThe record is now
on the common time grid
```

```
Out[9]:
```



In [10]:

```
p_title_preMPT
```

Out[10]:

pre-MPT

In [11]:

```
p1 = plot(p_La2004_preVSpost, title= "",  xlabel = "", xtickfont = false)
p2 = plot(p_LR04_preVSpost, xlabel = "", xtickfont = false)
p3 = plot(p_E_preVSpost, xlabel = "", xtickfont = false)
p4 = plot(p_MG_preVSpost,  xlabel = "", xtickfont = false)


p_xaxis_preMPT =
plot(title = "pre-MPT",
    xlabel =  "[kyrs BP]",  xlims = (gridstart_preMPT-5, gridend_preMPT+5),
    xticks = (gridstart_preMPT : gridend_preMPT - gridstart_preMPT : gridend_pre
MPT),
    ylims = (0,0), ylabel = "", yaxis = :off, grid = :off)

p_xaxis_postMPT =
plot(title = "post-MPT",
    xlabel =  "[kyrs BP]",  xlims = (gridstart_postMPT-5, gridend_postMPT+5),
    xticks = (gridstart_postMPT : gridend_postMPT - gridstart_postMPT : gridend_
postMPT),
    ylims = (0,0), ylabel = "",  yaxis = :off, grid = :off)

l = @layout[a b{0.6w}]
p_xaxis_preVSpost = plot(p_xaxis_preMPT, p_xaxis_postMPT, layout = l, xtickfont
= 8, xaxis = :true,)



l = @layout [a{0.011h}; b; c; d; e]
pe_timeseries =
plot(p_xaxis_preVSpost,
    p1,p2,p3,p4,
    layout = l,
    size = A4, #(400,600),
    #ymirror = false,
    #ylims = (-10,10),
    #yticks = (-20:5:20),
    legend = false,
    ymirror = true,

    )

plot(pe_timeseries,)
```

```
UndefVarError: plot not defined

Stacktrace:
 [1] top-level scope at In[11]:1
```

2. Setting the parameters for analyses

2.1 Data quality as time series length.

In this method, the data quality, depends on the length of the time series and their resolution. Since we have chosen the same records for the post- and pre-MPT time interval, and the records have fairly similar resolution over the two time windows (se discussion on resolution in NB1), it is mainly the time series length that determines the data quality. We will manipulate the post-MPT data quality by picking random sequences of pre-MPT length (325 data points) when computing the predictive asymmetries.

In [15]:

```
N_preMPT = length(bmpArray_preMPT)
N_postMPT = length(bmpArray_postMPT)

print("The post-MPT time window is N=", N_postMPT, " datapoints long, while
the pre-MPT time window is N=", N_preMPT, " datapoints long.
We will use ", N_preMPT, " as the length of the random sequences in computing th
e predictive asymmetries for both time windows")
```

```
The post-MPT time window is N=479 datapoints long, while
the pre-MPT time window is N=325 datapoints long.
We will use 325 as the length of the random sequences in computing t
he predictive asymmetries for both time windows
```

In [16]:

```
# Palus horizon
N_preMPT = length(bmpArray_preMPT)
eD = 3
ϵ = Int(round( N_preMPT^(1/(eD+1)) ))
print("According to the Palus horizon, we should then choose a binning of state
 space of
ϵ=", ϵ, " to estimate transfer entropy for time series of ", N_preMPT, " observat
ions.")
```

```
According to the Palus horizon, we should then choose a binning of s
tate space of
ϵ=4 to estimate transfer entropy for time series of 325 observations.
```

In [17]:

```
N = length(bmpArray_preMPT)
f = 1  # For time series longer than 150 data points, we use the significance treshold set by f=1 in normalization
ϵ = Int(round(N^(1/(3+1)) ))                     # Binning of state space is set by the Palus horison
rs = RandomSequences(150, N_preMPT : N_preMPT)  # Length of random sequences is equal to the time series length of the preMPT window (325 data points)
η = 1 # time step between each data point

print("ϵ = ", ϵ, "
f = 1
rs = ", rs, "
time step η = ", η, " (1 kyr)")
```

```
ϵ = 4
f = 1
rs = RandomSequences(150, 325:325)
time step η = 1 (1 kyr)
```

Compute the predictive asymmetry between the time series

See method explained in NB2 and function defined in NB3

computePredictiveAsymmetries(X::Any, # source time series first, target time series next (uivD, or Array if no uncertainty associated) Y::Any; # target time series first, source time series next timestep = 1, # length of time step is 1 kyr ηmax = 20, ϵ = 4, # binning of state space (Palus approximation to optimal binning for time series of length N=325) rs = RandomSequences(150, N : N), # calculate PA for 150 random sequences of same length as the pre-MPT interval (N = 325 data points). filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/[...]_postMPT.jld2", # assign a filepath for where to save the results)

Outline for analyses

1. $\delta^{18}O$ **- insolation** (LR04 - Ins)
2. **GSL - insolation** (EldSL - Ins)
3. $\delta^{18}O$ **- dust** (LR04 - MarFe)
4. **GSL - dust** (EldSL - MarFe)
5. **dust - insolation** (MarFe - Ins)

LR04 - insolation

In [28]:

```
# Recall the time series on the common grid as X and Y
X = LR04_cut_postMPT
Y = La2004_insol_cut_postMPT

computePredictiveAsymmetries(X, Y,
    timestep = η,  # length of time step is 1 kyr
    ηmax = 20,
    ϵ = 4,  # Binning of state space for time series (Palus horizon for time ser
ies of length N=325).
    rs = RandomSequences(150, N : N), # calculate PA for 150 random sequences of
same length as the pre-MPT interval (N = 325 data points).
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_data
quality/LR04_La2004_postMPT.jld2",  # assign a filepath for where to save the re
sults
    )
```

```
Results are saved in the .jld2 file.
```

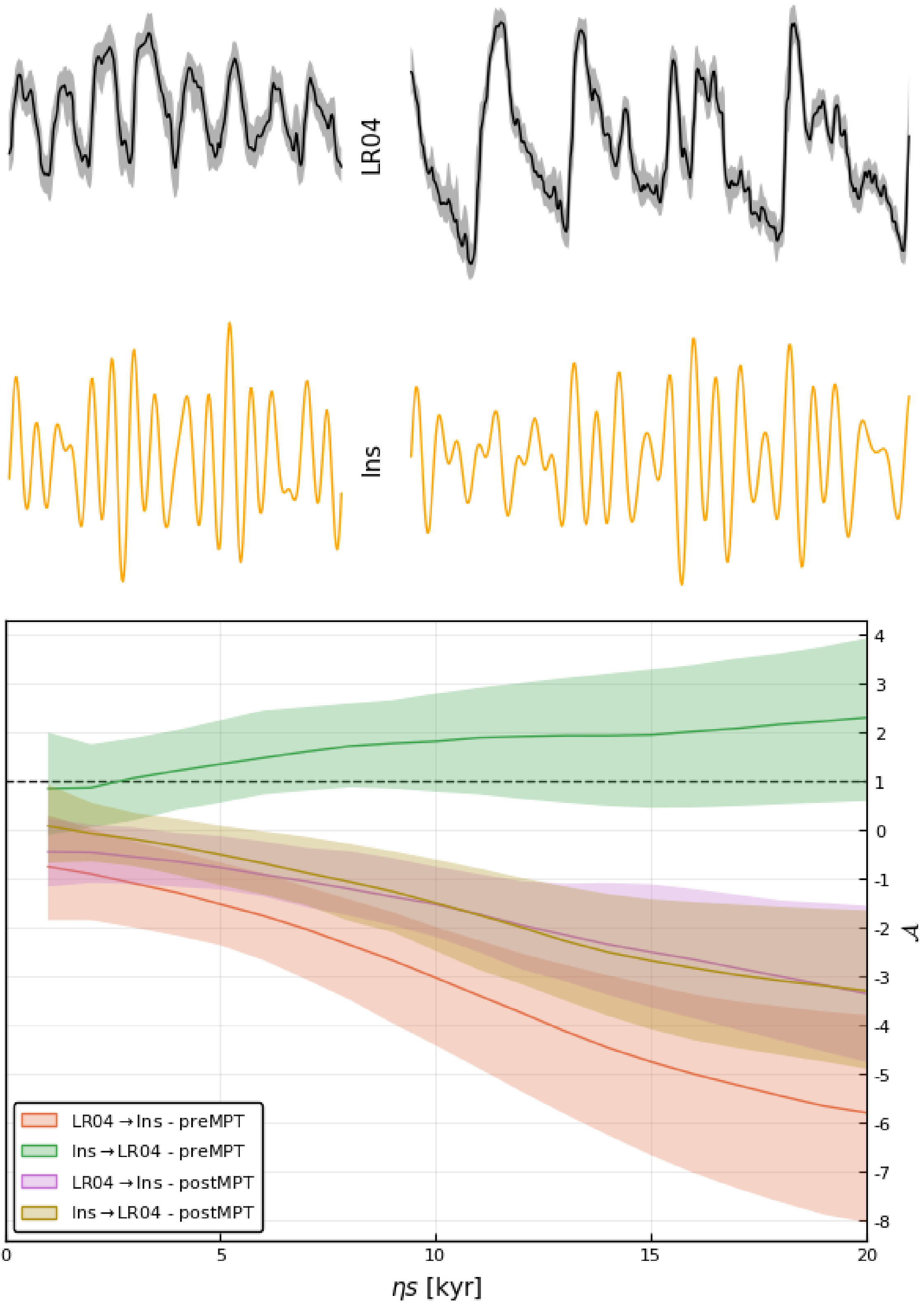
In [141]:

```
# Recall the time series on the common grid as X and Y
X = LR04_cut_preMPT
Y = La2004_insol_cut_preMPT

computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 4, rs = RandomSe
quences(150, N_preMPT : N_preMPT),
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_data
quality/LR04_La2004_preMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

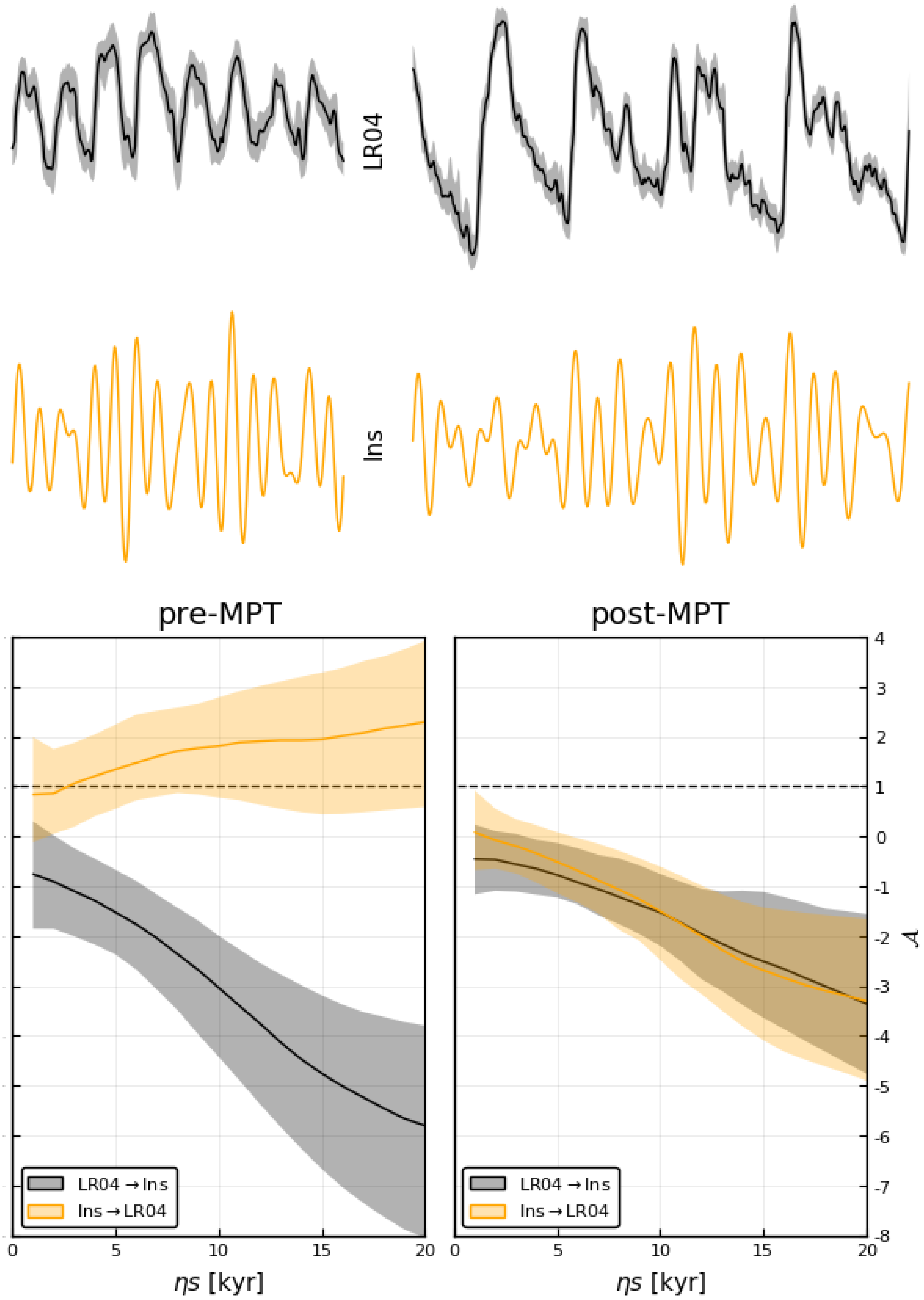
In [26]:

```
###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_La2004_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_La2004_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)



####### defining the results plot




# Plot the mean normalized predictive asymmetry with the 95% confidence interval
plot_normPA_LR04_La2004 =
plot(xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    yimrror = true,
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
# preMPT
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("LR04", L"\rightarrow", "Ins - preMPT"),
    fillalpha = 0.3, #color = :darkgrey
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("Ins", L"\rightarrow", "LR04 - preMPT"),
    fillalpha = 0.3, #color = :grey
    )
# post-MPT
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("LR04", L"\rightarrow", "Ins - postMPT"),
    fillalpha = 0.3, #color = :blue
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("Ins", L"\rightarrow", "LR04 - postMPT"),
    fillalpha = 0.3,# color = :blue
    )




########### defining the overview plot

plot_overview_LR04_La2004 =
plot(p_LR04_preVSpost, p_La2004_preVSpost, layout = (2,1))

# join plots of time series and pa results to a results subplot
plot_results_LR04_La2004 =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    ymirror = true,
    plot_overview_LR04_La2004,
    plot_normPA_LR04_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataqualit
y/LR04_La2004.pdf")
```


LR04
Ins
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
$\mathcal{A}$
0
5
10
15
20
ηs [kyr]
🟥 LR04 → Ins - preMPT
🟩 Ins → LR04 - preMPT
🟪 LR04 → Ins - postMPT
🟨 Ins → LR04 - postMPT

In [31]:

```
################### 2 plots - LR04-Ins

###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_La2004_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_La2004_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)




# Plot the mean normalized predictive asymmetry with the 95% confidence interval

# preMPT
plot_normPA_LR04_La2004_preMPT =
plot(title = "pre-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("LR04", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :black,
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("Ins", L"\rightarrow", "LR04"),
    fillalpha = 0.3, color = :orange,
    ylabel = "", ytickfont = false,
    )


# post-MPT
plot_normPA_LR04_La2004_postMPT =
plot(title = "post-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("LR04", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :black,
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("Ins", L"\rightarrow", "LR04"),
    fillalpha = 0.3, color = :orange,
    ylabel =  L"\mathcal{A}", ytickfont = 8, ymirror = true
    )


# defining the overview plot

l = @layout [a b]
plot_normPA_LR04_La2004_2panels =
plot(plot_normPA_LR04_La2004_preMPT, plot_normPA_LR04_La2004_postMPT,
    layout = l, size = (800,400),
    ylims = (-8,4), yticks = (-10:1:10),
    xlims = (0,ηmax), xticks = (0:5:ηmax)
)

plot_results_LR04_La2004 =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_LR04_La2004,
    plot_normPA_LR04_La2004_2panels)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataqualit
y/LR04_La2004_2panels.pdf")
```


LR04
Ins
pre-MPT
post-MPT
LR04 → Ins
Ins → LR04
LR04 → Ins
Ins → LR04
0
5
10
15
20
0
5
10
15
20
ηs [kyr]
ηs [kyr]
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
$\mathcal{A}$

2. $\mathcal{A}$ between GSL and insolation

Elderfield - La2004

In [1]:

```
# Recall the time series on the common grid as X and Y
X = E_cut_postMPT
Y = La2004_insol_cut_postMPT

computePredictiveAsymmetries(X, Y, timestep = η, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E_La2004_postMPT.jld2")
```

```
UndefVarError: E_cut_postMPT not defined

Stacktrace:
 [1] top-level scope at In[1]:1
```

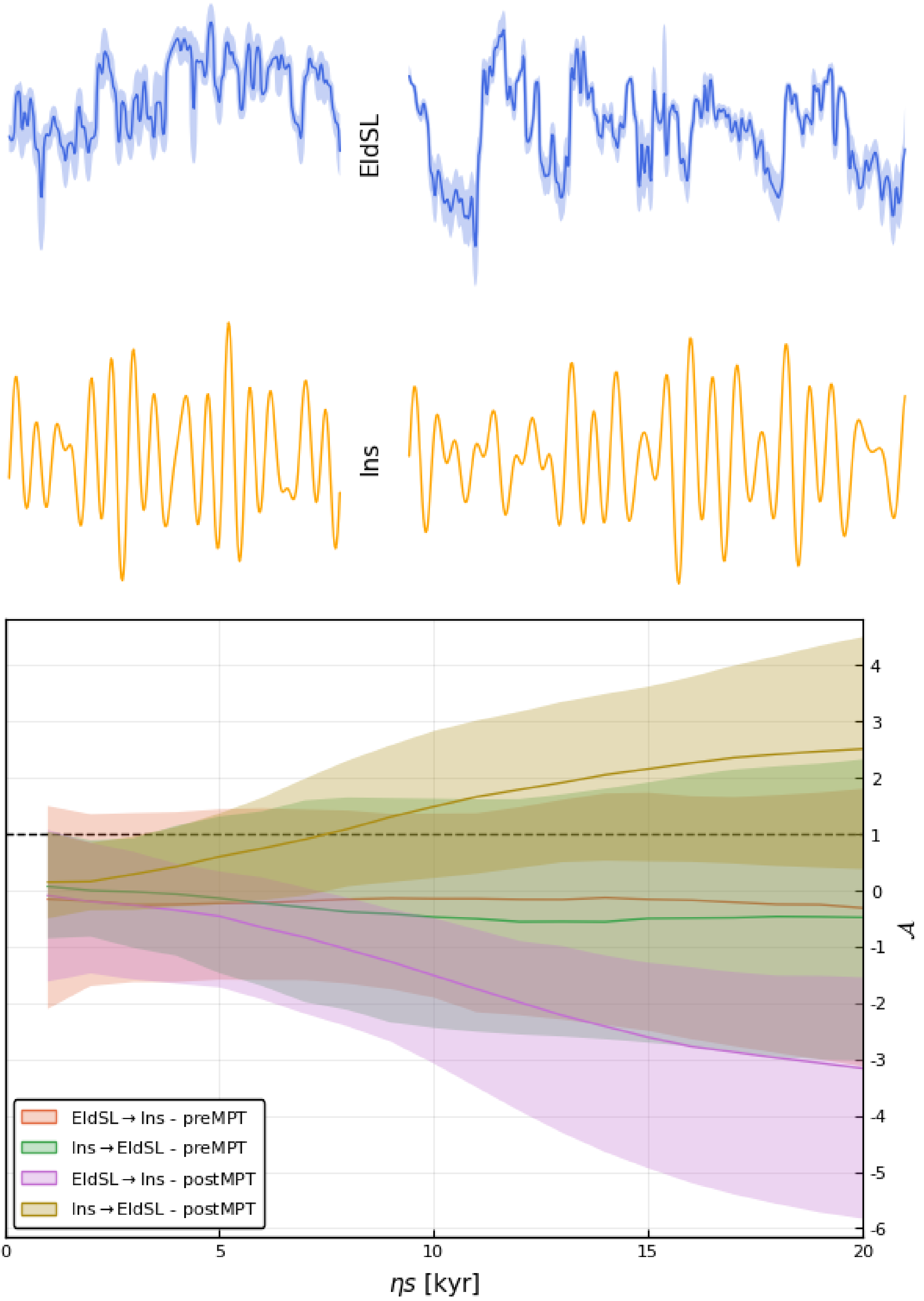
In [167]:

```
# Recall the time series on the common grid as X and Y
X = E_cut_preMPT
Y = La2004_insol_cut_preMPT

computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E_La2004_preMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

In [32]:

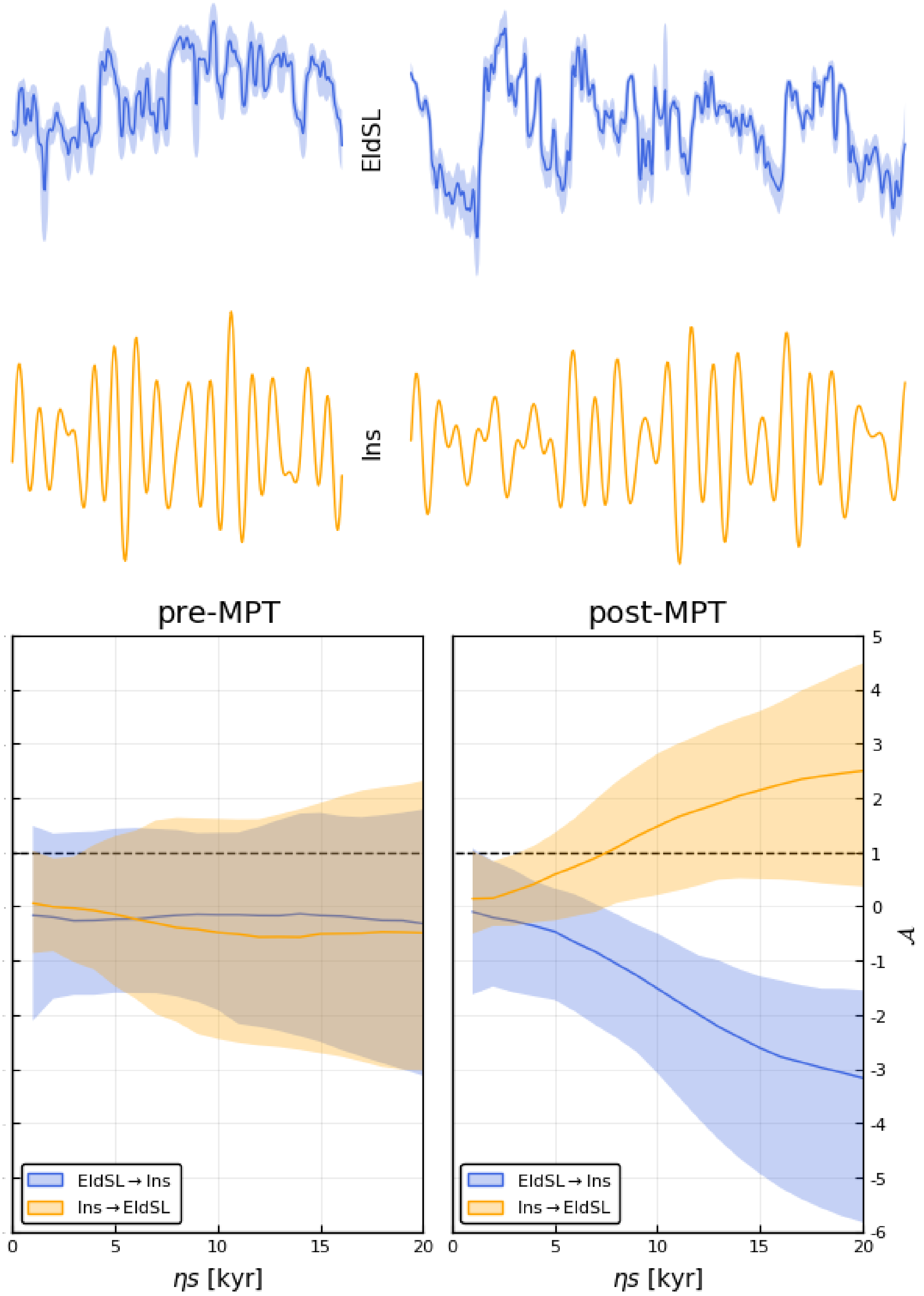
```
###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E
_La2004_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)



###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E
_La2004_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)




####### defining the results plot





# Plot the mean normalized predictive asymmetry with the 95% confidence interval
plot_normPA_E_La2004 =
plot(xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    ymirror = true,
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
# preMPT
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("EldSL", L"\rightarrow", "Ins - preMPT"),
    fillalpha = 0.3, # color = :darkgrey
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("Ins", L"\rightarrow", "EldSL - preMPT"),
    fillalpha = 0.3, #color = :grey
    )
# post-MPT
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("EldSL", L"\rightarrow", "Ins - postMPT"),
    fillalpha = 0.3, #color = :blue
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("Ins", L"\rightarrow", "EldSL - postMPT"),
    fillalpha = 0.3, # color = :blue
    )




########### defining the overview plot

plot_overview_E_La2004 =
plot(p_E_preVSpost, p_La2004_preVSpost, layout = (2,1))

# join plots of time series and pa results to a results subplot
plot_results_E_La2004 =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_E_La2004,
    plot_normPA_E_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataqualit
y/E_La2004.pdf")
```


EldSL
Ins
EldSL → Ins - preMPT
Ins → EldSL - preMPT
EldSL → Ins - postMPT
Ins → EldSL - postMPT
0
5
10
15
20
ηs [kyr]
4
3
2
1
0
-1
-2
-3
-4
-5
-6
𝒜

In [33]:

```
################### 2 plots - EldSL-Ins

###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E
_La2004_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)
# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E
_La2004_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)
# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)




# Plot the mean normalized predictive rate asymmetry with the 95% confidence interval

# preMPT
plot_normPA_E_La2004_preMPT =
plot(title = "pre-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
```

```
    )
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("EldSL", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :royalblue,
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("Ins", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, color = :orange,
    ylabel = "", ytickfont = false,
    )


# post-MPT
plot_normPA_E_La2004_postMPT =
plot(title = "post-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance treshold)
    )
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("EldSL", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :royalblue,
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("Ins", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, color = :orange,
    ylabel =  L"\mathcal{A}", ytickfont = 8, ymirror = true
    )


# defining the overview plot

l = @layout [a b]
plot_normPA_E_La2004_2panels =
plot(plot_normPA_E_La2004_preMPT, plot_normPA_E_La2004_postMPT,
    layout = l, size = (800,400),
    ylims = (-6,5), yticks = (-10:1:10),
    xlims = (0,ηmax), xticks = (0:5:ηmax)
)

plot_results_E_La2004 =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_E_La2004,
    plot_normPA_E_La2004_2panels)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataquality/E_La2004_2panels.pdf")
```



3. $\mathcal{A}$ between GSL - Fe flux

Elderfield - Martinez-García

In [56]:

```
# Recall the time series on the common grid as X and Y
X = E_cut_preMPT
Y = MG_cut_preMPT

computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_data
quality/E_MG_preMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

In [19]:

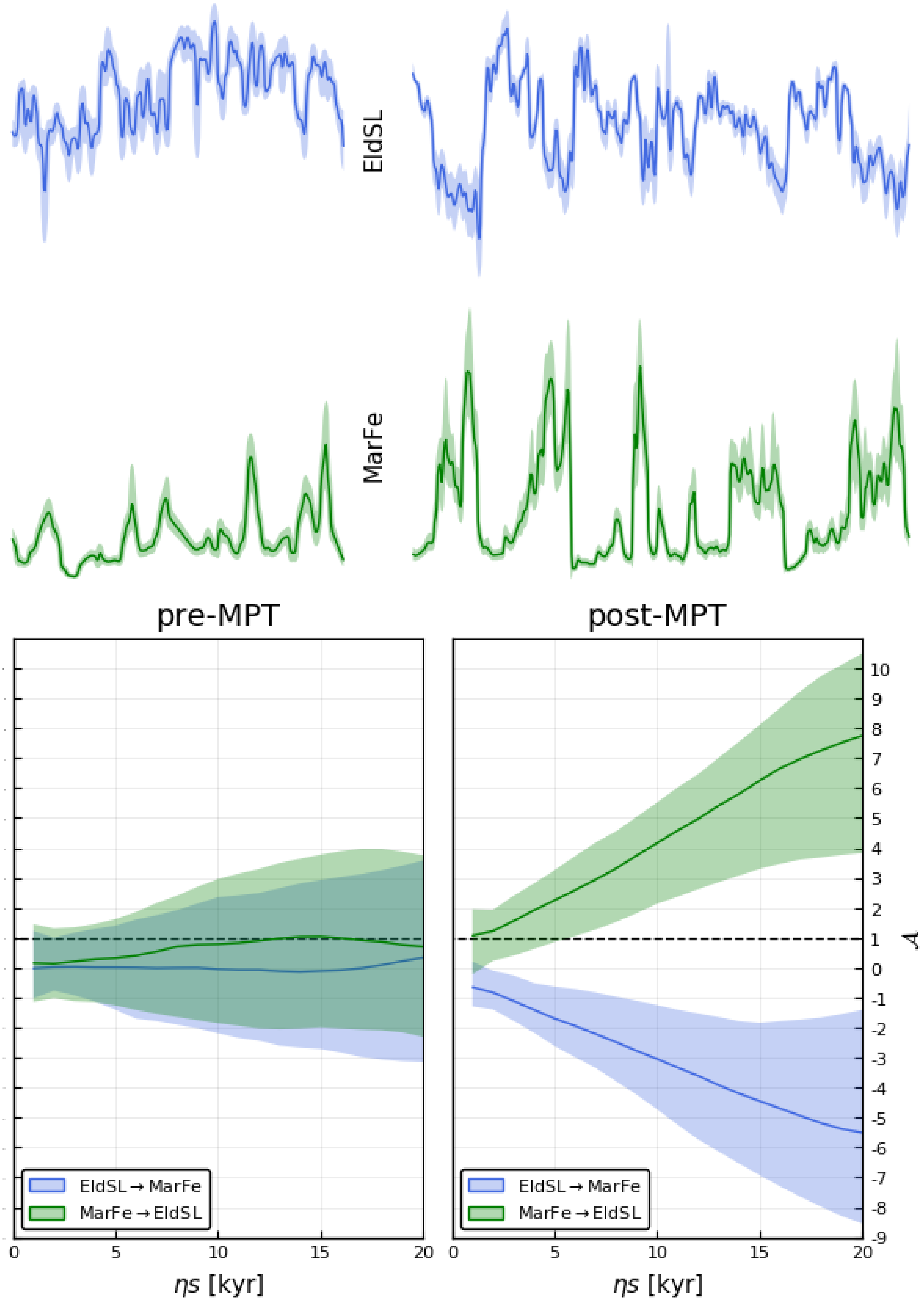
```
# Recall the time series on the common grid as X and Y
X = E_cut_postMPT
Y = MG_cut_postMPT

computePredictiveAsymmetries(X, Y, timestep = η, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_data
quality/E_MG_postMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

In [21]:

```
###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E
_MG_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,    # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,    # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )       # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/E
_MG_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)



####### defining the results plot




# Plot the mean normalized predictive asymmetry with the 95% confidence interval

plot_normPA_E_MG =
plot(xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance treshold)
    )
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("EldSL", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, #color = :royalblue,
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("MarFe", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, #color = :green,
    )
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("EldSL", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, #color = :royalblue,
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("MarFe", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, #color = #:green,
    ylabel = "", ytickfont = false
    )

# defining the overview plot

plot_overview_E_MG =
plot(p_E_preVSpost, p_MG_preVSpost, layout = (2,1))

# join plots of time series and pa results to a results subplot

plot_results_E_MG =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_E_MG,
    plot_normPA_E_MG)

#savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataquality/E_MG.pdf")




################### 2 plots
# Plot the mean normalized predictive asymmetry with the 95% confidence interval

# preMPT
plot_normPA_E_MG_preMPT =
plot(title = "pre-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance treshold)
    )
```

```
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("EldSL", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :royalblue,
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("MarFe", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, color = :green,
    ylabel = "", ytickfont = false,
    )

# post-MPT
plot_normPA_E_MG_postMPT =
plot(title = "post-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("EldSL", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :royalblue,
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("MarFe", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, color = :green,
    ymirror = true
)


# defining the overview plot

l = @layout [a b]
plot_normPA_E_MG_2panels =
plot(plot_normPA_E_MG_preMPT, plot_normPA_E_MG_postMPT,
    layout = l, size = (800,400),
    ylims = (-9,11), yticks = (-10:1:10),
    xlims = (0,ηmax), xticks = (0:5:ηmax)
)

plot_results_E_MG =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_E_MG,
    plot_normPA_E_MG_2panels)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataqualit
y/E_MG_2panels.pdf")
```


EldSL
MarFe
pre-MPT
post-MPT
EldSL → MarFe
MarFe → EldSL
EldSL → MarFe
MarFe → EldSL
0
5
10
15
20
ηs [kyr]
0
5
10
15
20
ηs [kyr]
10
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
$\mathcal{A}$

LR04 - Martinez-García

In [29]:

```
# Recall the time series on the common grid as X and Y
X = LR04_cut_preMPT
Y = MG_cut_preMPT

computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/LR04_MG_preMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

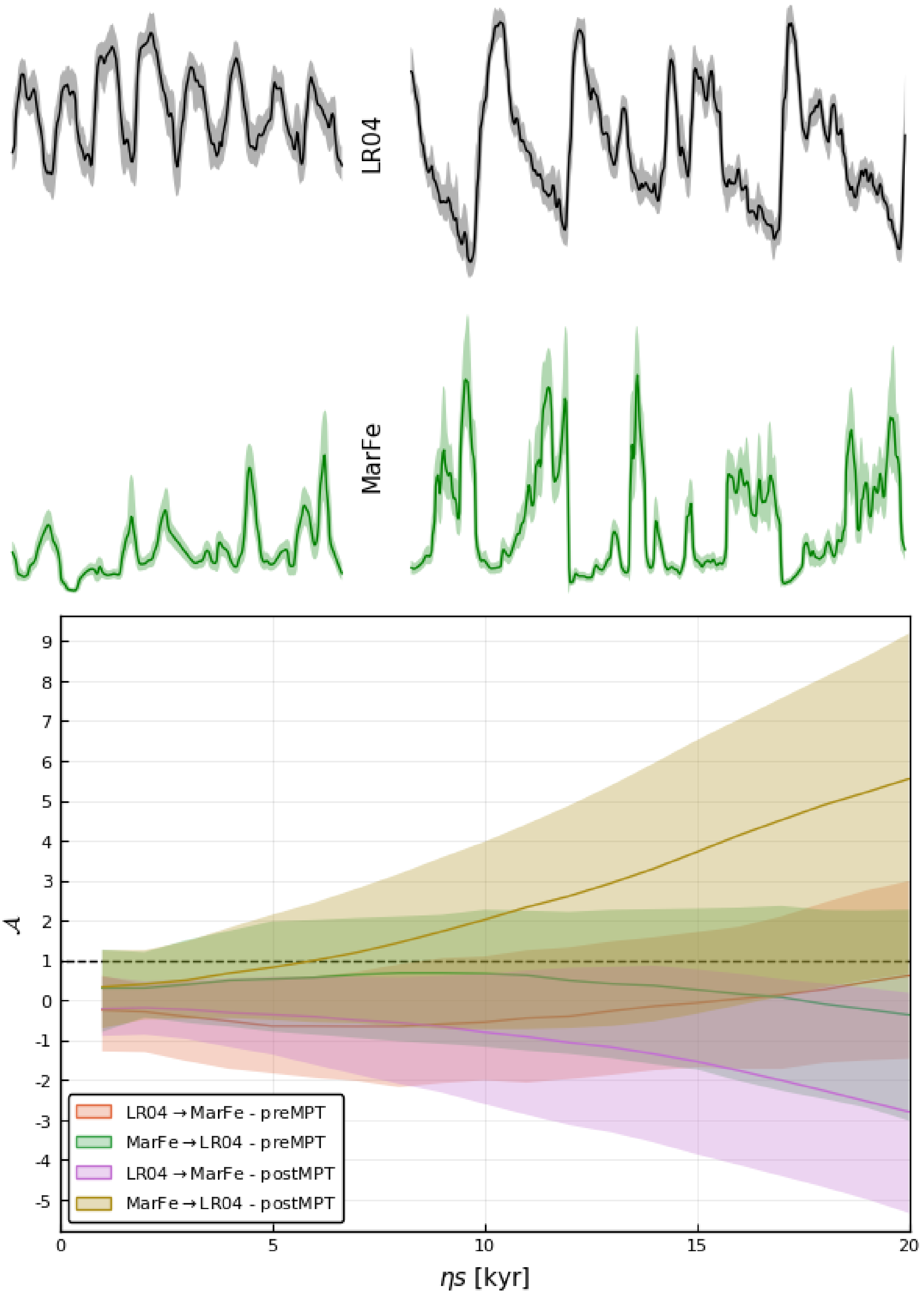
In [30]:

```
# Recall the time series on the common grid as X and Y
X = LR04_cut_postMPT
Y = MG_cut_postMPT

computePredictiveAsymmetries(X, Y, timestep = η, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/LR04_MG_postMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

In [36]:

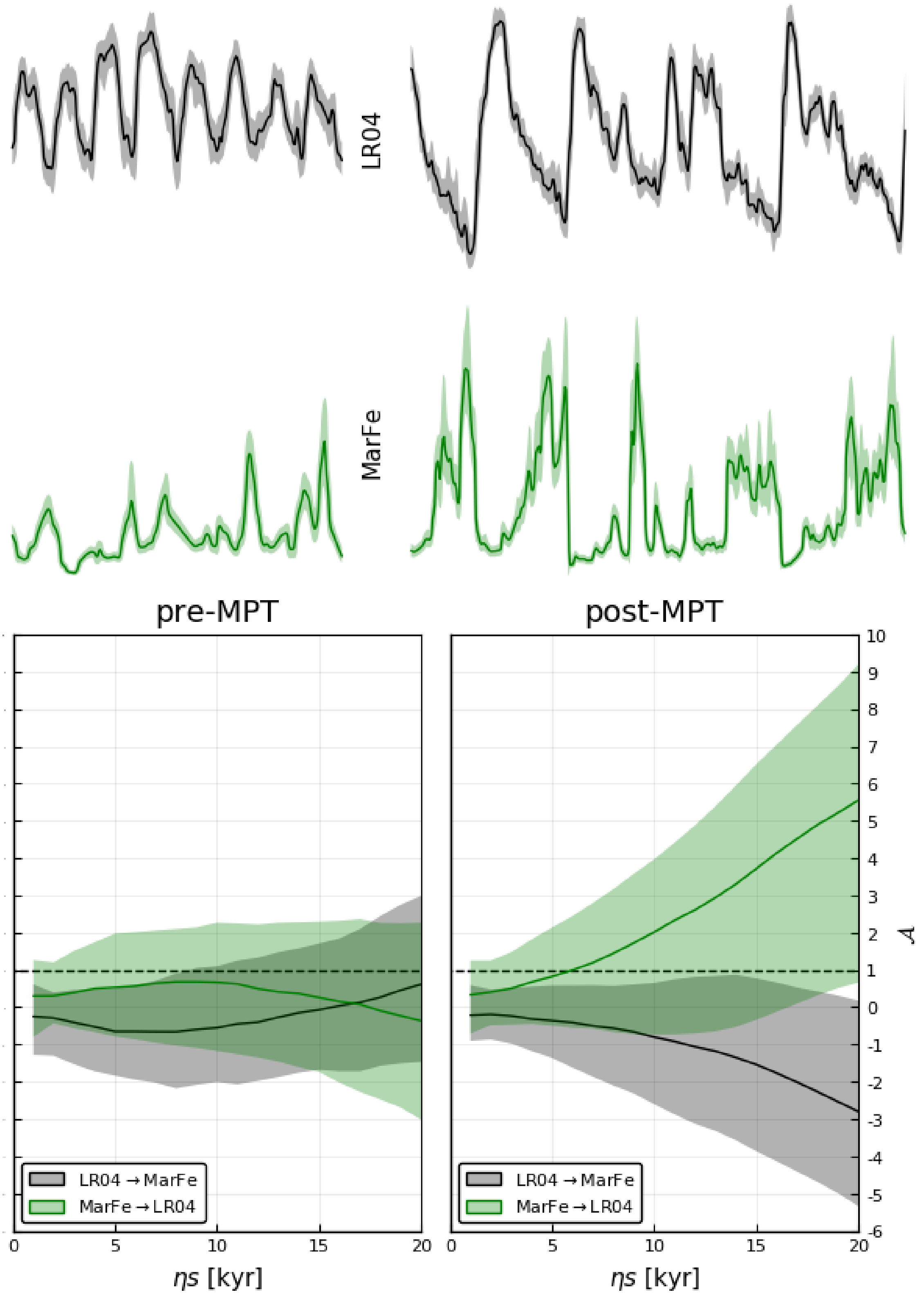
```
###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_MG_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_MG_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)



####### defining the results plot




# Plot the mean normalized predictive asymmetry with the 95% confidence interval
plot_normPA_LR04_MG =
plot(xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
# preMPT
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("LR04", L"\rightarrow", "MarFe - preMPT"),
    fillalpha = 0.3, #color = :lightblue
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("MarFe", L"\rightarrow", "LR04 - preMPT"),
    fillalpha = 0.3, #color = :lightgreen
    )
# post-MPT
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("LR04", L"\rightarrow", "MarFe - postMPT"),
    fillalpha = 0.3, #color = :darkblue
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("MarFe", L"\rightarrow", "LR04 - postMPT"),
    fillalpha = 0.3,  #color = :darkgreen
    )




########### defining the overview plot

plot_overview_LR04_MG =
plot(p_LR04_preVSpost, p_MG_preVSpost, layout = (2,1))

# join plots of time series and pa results to a results subplot
plot_results_LR04_MG =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_LR04_MG,
    plot_normPA_LR04_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataquality/LR04_MG.pdf") ;
```


LR04
MarFe
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
$\mathcal{A}$
0
5
10
15
20
ηs [kyr]
LR04 → MarFe - preMPT
MarFe → LR04 - preMPT
LR04 → MarFe - postMPT
MarFe → LR04 - postMPT

In [37]:

```
########################################  plot results in 2 panels - LR04-MarFe

###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_MG_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
R04_MG_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)




# preMPT
plot_normPA_LR04_MG_preMPT =
plot(title = "pre-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
```

```
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("LR04", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :black
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("MarFe", L"\rightarrow", "LR04"),
    fillalpha = 0.3, color = :green,
    ylabel = "", ytickfont = false
    )



# post-MPT
plot_normPA_LR04_MG_postMPT =
plot(title = "post-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("LR04", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :black
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("MarFe", L"\rightarrow", "LR04"),
    fillalpha = 0.3,  color = :green,
    ymirror = true,
    )



l = @layout [a b]
plot_normPA_LR04_MG_2panels =
plot(plot_normPA_LR04_MG_preMPT, plot_normPA_LR04_MG_postMPT,
    layout = l, size = (800,400),
    ylims = (-6,10), yticks = (-10:1:10),
    xlims = (0,ηmax), xticks = (0:5:ηmax)
)

plot_results_LR04_MG_2panels =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_LR04_MG,
    plot_normPA_LR04_MG_2panels)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataqualit
y/LR04_MG_2panels.pdf")
```



$\mathcal{A}$ between northern hemsiphere summer insolation and Southern hemisphere aeolian dust

La2004 - Martinez-García

In [204]:

```
# Recall the time series on the common grid as X and Y
X = La2004_insol_cut_preMPT
Y = MG_cut_preMPT

computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/La2004_MG_preMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

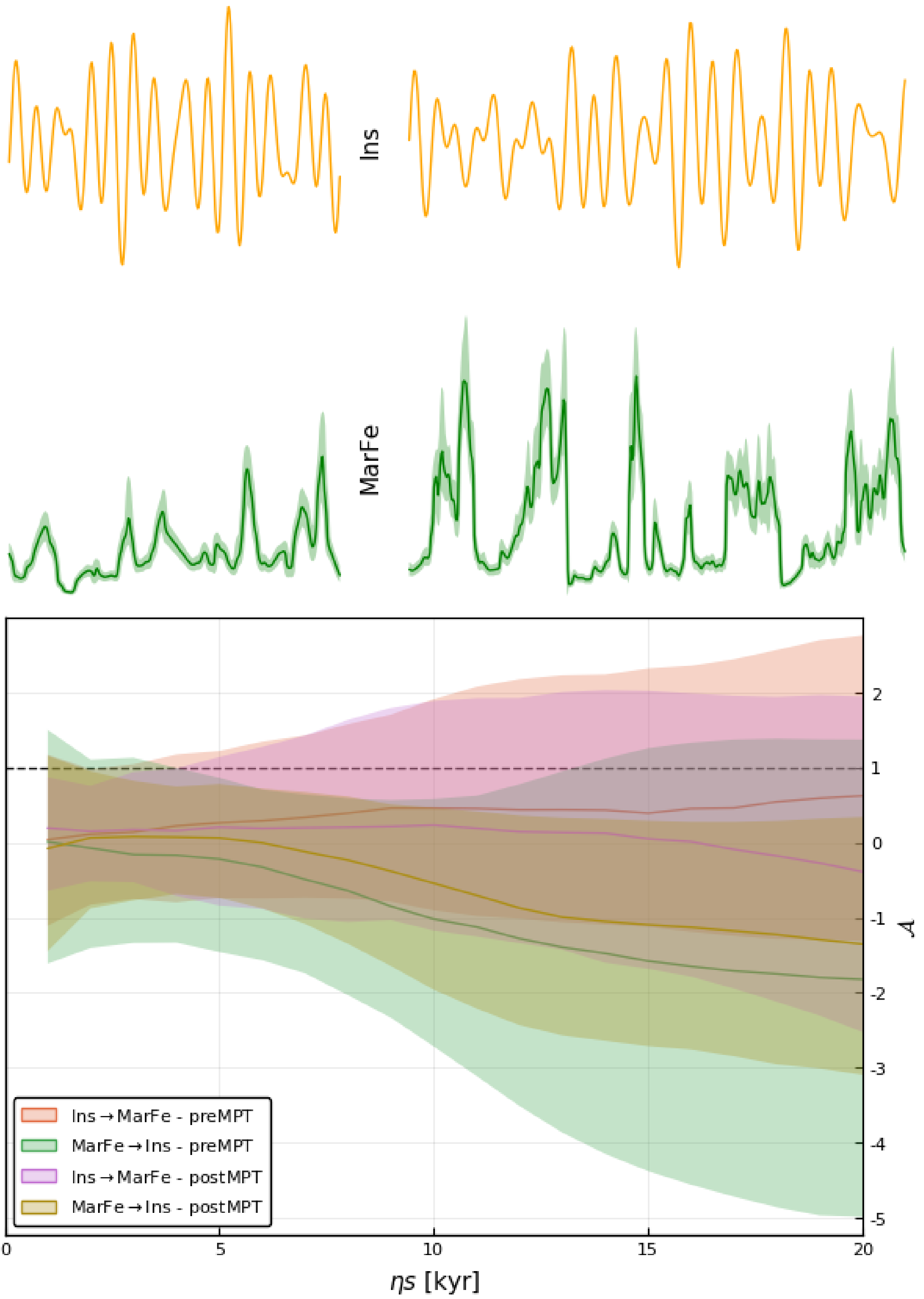
In [206]:

```
# Recall the time series on the common grid as X and Y
X = La2004_insol_cut_postMPT
Y = MG_cut_postMPT

computePredictiveAsymmetries(X, Y, timestep = η, ηmax = 20, ϵ = ϵ, rs = rs,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/La2004_MG_postMPT.jld2")
```

```
Results are saved in the .jld2 file.
```

In [22]:

```
###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
a2004_MG_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequenc
es
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200
 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT =
quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT =
quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax
= ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/L
a2004_MG_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT
= quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT
= quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηm
ax = ηmax)



####### defining the results plot




# Plot the mean normalized predictive asymmetry with the 95% confidence interval
plot_normPA_La2004_MG =
plot(xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    ymirror = true,
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
# preMPT
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("Ins", L"\rightarrow", "MarFe - preMPT"),
    fillalpha = 0.3, # color = :darkgrey
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("MarFe", L"\rightarrow", "Ins - preMPT"),
    fillalpha = 0.3, #color = :grey
    )
# post-MPT
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("Ins", L"\rightarrow", "MarFe - postMPT"),
    fillalpha = 0.3, #color = :blue
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("MarFe", L"\rightarrow", "Ins - postMPT"),
    fillalpha = 0.3, # color = :blue
    )




########### defining the overview plot

plot_overview_La2004_MG =
plot(p_La2004_preVSpost, p_MG_preVSpost, layout = (2,1))

# join plots of time series and pa results to a results subplot
plot_results_La2004_MG =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_La2004_MG,
    plot_normPA_La2004_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataqualit
y/La2004_MG.pdf")
```

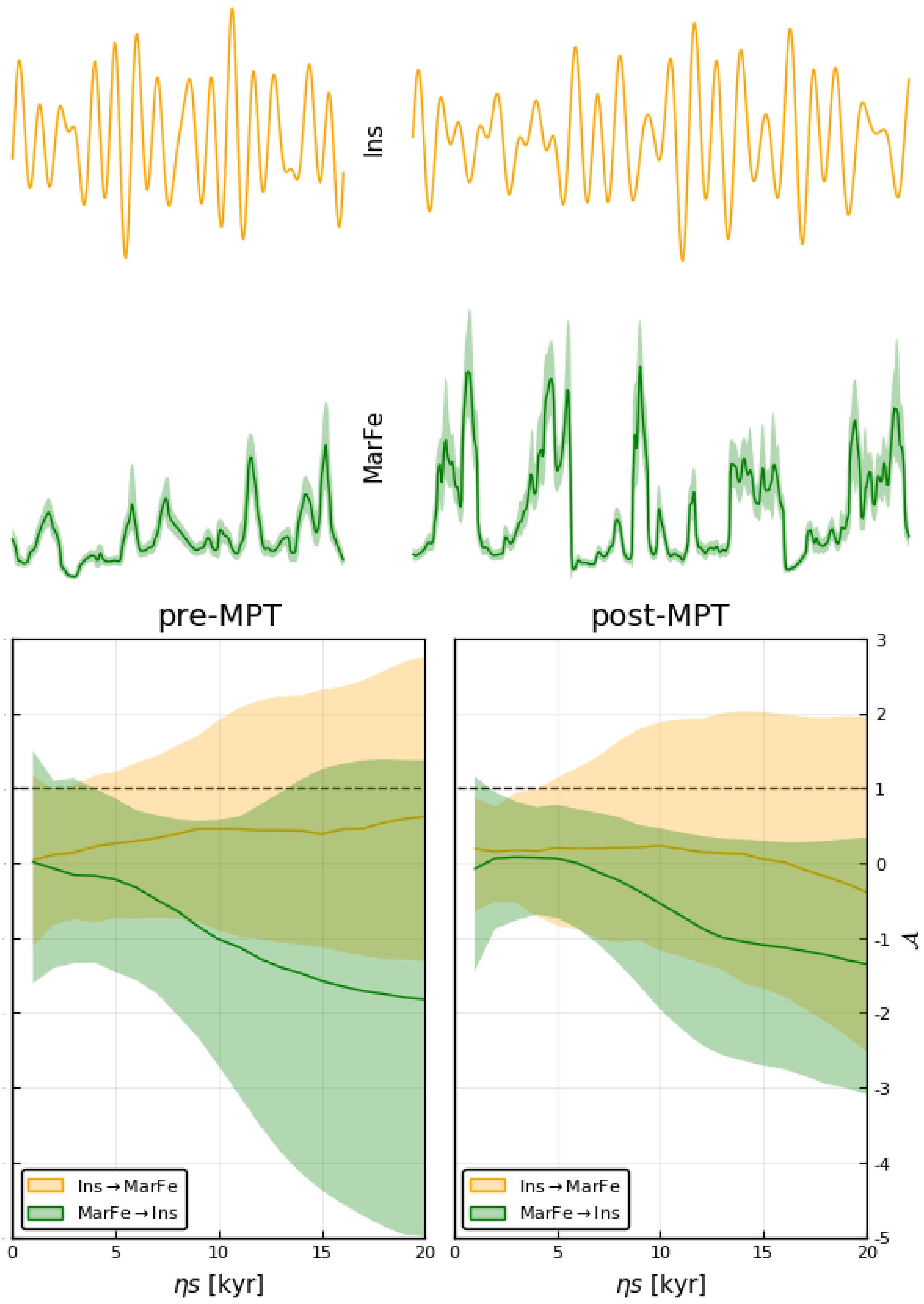


- results plot with 2 panels

In [28]:

```
########################################  plot results in 2 panels -- Ins-MarFe


###### normalize the pre-MPT results

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/La2004_MG_preMPT.jld2"

normPA_XtoY = normalizePredictiveAsymmetry(
    rsTE_XtoY,   # arrays of transfer entropy for a family of random sequences
    rsPA_XtoY,   # arrays of predictive asymmetry for a family of random sequences
    ηmax = ηmax, # 20 prediction lags
    f = 1 )      # false positive rate set to 1 for time series longer than 200 data points (see Haaga et al.).

normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f = 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_preMPT, normPA_XtoY_upper_preMPT, normPA_XtoY_lower_preMPT = quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax = ηmax)
normPA_YtoX_median_preMPT, normPA_YtoX_upper_preMPT, normPA_YtoX_lower_preMPT = quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax = ηmax)


###### post-MPT

@load "../../results_ePalus_ns20/pa_jld2_files/postMPT_with_preMPT_dataquality/La2004_MG_postMPT.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f = 1 )
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f = 1)

# calculate the quantiles for the 95% confidence interval
normPA_XtoY_median_postMPT, normPA_XtoY_upper_postMPT, normPA_XtoY_lower_postMPT = quantiles_normPA(normPA_XtoY, upperquantile = 0.975, lowerquantile = 0.025, ηmax = ηmax)
normPA_YtoX_median_postMPT, normPA_YtoX_upper_postMPT, normPA_YtoX_lower_postMPT = quantiles_normPA(normPA_YtoX, upperquantile = 0.975, lowerquantile = 0.025, ηmax = ηmax)


##########

# preMPT
plot_normPA_La2004_MG_preMPT =
plot(title = "pre-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
plot!(normPA_XtoY_median_preMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_preMPT, normPA_XtoY_upper_preMPT),
    label =  string("Ins", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median_preMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_preMPT, normPA_YtoX_upper_preMPT),
    label =  string("MarFe", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :green,
    ylabel = "", ytickfont = false
    )



# post-MPT
plot_normPA_La2004_MG_postMPT =
plot(title = "post-MPT",
    xlims = (0, ηmax), xticks = (0 : 5 : ηmax),
    #ylims = (-3,3),
    yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), grid = true, border = true,
    hline([1], line = (:dash, :black)), label =  "" # mean TE (significance tres
hold)
    )
plot!(normPA_XtoY_median_postMPT, # ...from X to Y
    ribbon = (normPA_XtoY_lower_postMPT, normPA_XtoY_upper_postMPT),
    label =  string("Ins", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median_postMPT, # ...from Y to X
    ribbon = (normPA_YtoX_lower_postMPT, normPA_YtoX_upper_postMPT),
    label =  string("MarFe", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :green,
    ymirror = true,
    )

########### defining the overview plot

plot_overview_La2004_MG =
plot(p_La2004_preVSpost, p_MG_preVSpost, layout = (2,1))


l = @layout [a b]
plot_normPA_La2004_MG_2panels =
plot(plot_normPA_La2004_MG_preMPT, plot_normPA_La2004_MG_postMPT,
    layout = l, size = (800,400),
    ylims = (-5,3), yticks = (-10:1:10),
    xlims = (0,ηmax), xticks = (0:5:ηmax)
)

plot_results_La2004_MG_2panels =
plot(size = A4, #(400,1000),
    layout = grid(2,1),
    bg_legend = :white,
    plot_overview_La2004_MG,
    plot_normPA_La2004_MG_2panels)
```

```
savefig("../../results_ePalus_ns20/pa_ResultPlots/postMPT_with_preMPT_dataqualit
y/La2004_MG_2panels.pdf")
```



Ensemble plots of results

In [24]:

```
panel_labels = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m",
"n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"];
```

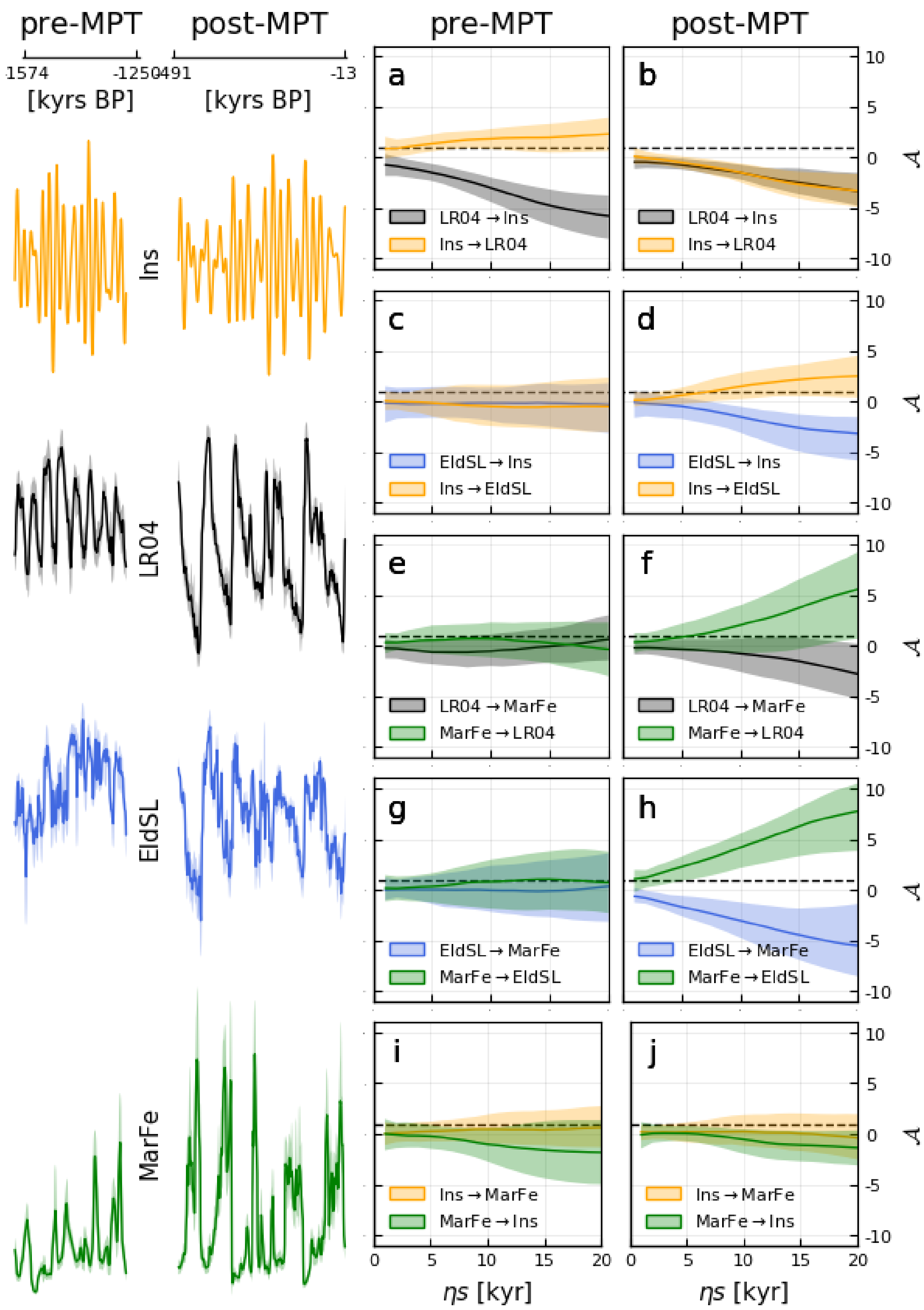
In [ ]:

In [41]:

```
#plot(pe_timeseries, xaxis = :off, yaxis = :off, xlabel = "", legend = :outerright, size = (400,600), ylabel = "", bg_legend = :transparent, ymirror = false)
#plot(pe_timeseries, xaxis = :off, yaxis = :off, xlabel = "")
########### results in 1 panel

p1 = plot(plot_normPA_LR04_La2004, ylabel = string(L"\delta^{18}O", " - insolation"), xtickfont = false, xlabel = "", ytickfont = false, title = "post- VS pre-MPT")
p2 = plot(plot_normPA_E_La2004, ylabel = string("GSL", " - insolation"), xtickfont = false, xlabel = "", ytickfont = false)
p3 = plot(plot_normPA_LR04_MG, ylabel = string(L"\delta^{18}O", " - Fe flux"), xtickfont = false, xlabel = "", ytickfont = false)
p4 = plot(plot_normPA_E_MG, ylabel = string("GSL", " - Fe flux"), xtickfont = false, xlabel = "", ytickfont = false)
p5 = plot(plot_normPA_La2004_MG, ylabel = string("Insolation - Fe flux"), ytickfont = false, xlabel = "ηs [kyr]")

l = @layout [a; b; c; d; e]
pe_allresults_1panel =
plot(p1,p2,p3,p4, p5,
    layout = l,
    size = A4, #(500,5*250),
    ylims = (-11,11),
    yticks = (-20:5:20),
    ymirror = false
    )

############# results in 2 panels
p1 = plot(plot_normPA_LR04_La2004_2panels,   xlabel = "", xtickfont = false)
p2 = plot(plot_normPA_E_La2004_2panels,      title = "", xlabel = "", xtickfont = false)
p3 = plot(plot_normPA_LR04_MG_2panels,       title = "", xlabel = "", xtickfont = false)
p4 = plot(plot_normPA_E_MG_2panels,          title = "", xlabel = "", xtickfont = false)
p5 = plot(plot_normPA_La2004_MG_2panels,     title = "",)

l = @layout [a; b; c; d; e]
pe_allresults_2panels =
plot(p1,p2,p3,p4, p5,
    layout = l,
    size = (500,5*250),
    ylims = (-11,11),
    yticks = (-20:5:20),
    legend = :bottomleft
    )

for i in 1:12
    annotate!(2,8, panel_labels[i], subplot = i)
end

####### Appendix plot with time series

######### results in 3 panels
#l= @layout [a b{0.6w}]
#pe_allresults_3panels = plot(pe_allresults_1panel, pe_allresults_2panels, layout = l, size = A4, bg_legend = :transparent)
#l = @layout [a{0.2h}; b ]
#plot(pe_timeseries, pe_allresults_3panels, layout = l, size = A4)
```

```
l= @layout [a b{0.6w}]
pe_Appendix = plot(pe_timeseries, pe_allresults_2panels, layout = l, size = A4,
)
```

```
Out[41]:
```


pre-MPT
post-MPT
-1574
-1250
-491
-13
[kyrs BP]
[kyrs BP]
Ins
LR04
EldSL
MarFe
pre-MPT
post-MPT
a
LR04 → Ins
Ins → LR04
b
LR04 → Ins
Ins → LR04
c
EldSL → Ins
Ins → EldSL
d
EldSL → Ins
Ins → EldSL
e
LR04 → MarFe
MarFe → LR04
f
LR04 → MarFe
MarFe → LR04
g
EldSL → MarFe
MarFe → EldSL
h
EldSL → MarFe
MarFe → EldSL
i
Ins → MarFe
MarFe → Ins
j
Ins → MarFe
MarFe → Ins
10
5
0
-5
-10
$\mathcal{A}$
0
5
10
15
20
ηs [kyr]
ηs [kyr]

In [42]:

```
savefig("../../results_ePalus_ns20/e_allresults/Appendix_pre-VS-postMPT_wPL.pdf"
)
```

In [ ]:

```
```