time grid label for this notebook: `NBR_preMPT_1500`

Predictive asymmetry analysis of climate system dynamics prior to the MPT.

Written by Maria Salem, in supplement to master thesis.

In this notebook we compute the normalized predictive asymmetry ($\mathcal{A}$) between empirical climate proxy records over the last 320 kyrs leading up to the MPT (time interval 1570-1250 ka BP). The aim is to deduce causal coupling strength and directionality in the climate system between ice volume, northern hemisphere summer insolation, atmospheric pCO2 concentration, and Southern Ocean Fe fertilization prior to the Mid-Pleistocene Transition.

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" ;
```

```
The LR04 d18O record spans from -5319.0 to -1.0 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 -802.0 to -3.0 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.  41.2704
20 seconds (164.39 M allocations: 11.112 GiB, 14.34% gc time)
```

The following records are used for analysis (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)*.

Unfortunately, we do not have any proxy for pCO2 with satisfactory resolution over this time interval.

Outline for this notebook:

Get the time series on a common time grid

1. Decide on a common time interval for analyses in this notebook.
2. Cut all records to the relevant time interval.

Run pairwise analyses for predictive asymmetry between the time series

1. Compute a family of transfer entropies and predictive asymmetries for each time series pair (function `computePredictiveAsymmetry` defined in NB3).
2. Normalize the predictive asymmmetries and produce a results plot showing the 95% confidence interval of predictive asymmetry between each time series pair. **Produce an overview plot of predictive asymmetry results over the notebook's time interval**
3. Produce an overview plot of the ensemble of predictive asymmetry results for the time period here studied. The ensemble plot will be use in the results chapter of main text.

Get the time series on a common time grid

For our method to work, it is important we have the data on the same time grid - covering the exact same time interval and with a regular time step. The time step was set by binsize of the grid in the BinnedResampling in NB1, where we chose a regular grid with one value for every 1000 years. The common time interval, on the other hand, we set individually in each notebook.

1. Decide on common time grid

Determine the common time interval for analyses in this notebook. We decide on a time interval for analysis according to where the records overlap. Additionally, we can delimit the time interval to a period of interest - in our case, we want to see if there are any changes in causal dynamics before, during and after the Mid-Pleistocene Transition.

In [2]:

```
# Determine the common time interval

# the record with the "latest start" defines the start of the common grid
gridstart = maximum([
        tmin_LR04,
        tmin_La2004,
        tmin_E,
        tmin_R,
        tmin_MG
        ])

# the onset of the MPT defines the end of the common grid
gridend = -1250 # Onset of the MPT

print("Time interval for analyses in this notebook is from ", -gridstart, " ka B
P to ", -gridend, " ka BP")
# Note: opposite to NB1, tmin and tmax here indicate the binmidpoints of the com
mon grid.
```

```
Time interval for analyses in this notebook is from 1574.0 ka BP to
1250 ka BP
```

Recall the time step for the common grid. This was given by the binsize in the grids for BinnedResampling in NB1. We also recall time steps for the additional analyses of the high resolution records.

In [3]:

```
# Recall the binsize
binsize_1 = 1           # 1 kyrs - is the default timestep on which all time seri
es are binned (NB1)

print("All records are on a regular grid with timestep of ", binsize_1, " kyr.")
```

```
All records are on a regular grid with timestep of 1 kyr.
```

We can now define the time grid for analyses in this notebook. Objects associated with this grid are labeled with the grid suffix `preMPT_1500` .

In [4]:

```
# the common grids for the time series are then defined by

commongrid_1 = gridstart : 1 : gridend
Binmidpoints_commongrid_1 =[commongrid_1[i] for i in 1:length(commongrid_1)]

print(gridstart : 1 : gridend , " defines the main common grid for analyses in t
his notebook.")

#= *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.) =#
```

```
-1574.0:1.0:-1250.0 defines the main common grid for analyses in thi
s notebook.
```

Let's see what binning/graining of state space is used for estimations of transfer entropy in this notebook. The transfer entropy is a probability distribution, which we estimate by the visitation frequency test. Our estimation of the transfer entropy is thus sensitive to the binning resolution of state space (the amount of bins we use to count visitations), which is given by ϵ. ϵ is chosen under the hood of the `computePredictiveAsymmetry` function (see NB3), as a function of time series length. We use Milan Palus' [] proposed rule of thumb to approximate the more optimal binning of state space, which we refer to as the Palus horizon. As a curiosity, let's see what gridding of state space is for estimations of transfer entropy in this notebook. For overview, let's check the ϵ used for the time series in this notebook.

In [5]:

```
# Palus horizon

# 1kyr time step
N_1 = length(Binmidpoints_commongrid_1)
eD = 3 # embedding dimension in the VisitationFrequencyTest - we have used the d
efault 3
ϵ = Int(round( N_1^(1/(eD+1)) ))
print("The binning argument ϵ (defines the gridding of state space to estimate t
ransfer entropy) is
    ", ϵ, " for the 1 kyr timestep grid.")
```

```
The binning argument ϵ (defines the gridding of state space to estima
te transfer entropy) is
    4 for the 1 kyr timestep grid.
```

False positive rate. As shown by sensitivity tests by Haaga et al. (2020), the false positive rate *f* is a function of the time series length. Haaga et al. (2020) heuristically show $\mathcal{A}$ to have a significant bettering of the false positive rate at a time series length of around 150 datapoints, for most types of systems, with a few weird-ass exceptions such as for Henon-systems). We therefore choose to set our *f* to 1 (%?) for analyses with time series length above 200 datapoints (to be on the safer side). For time series of shorter length, we raise *f* to 1.5. A higher *f* means we will get fewer false positives (higher specificity), but also more false negatives (lower sensitivity). We make this choice to be sure we can draw robust conclusions.

In [6]:

```
# time series lengths in this notebook
print("Time series on the 1 kyr grid are of length ", length(Binmidpoints_common
grid_1), " - we will use f = 1 in interpretation of the results.")
```

```
Time series on the 1 kyr grid are of length 325 - we will use f = 1
in interpretation of the results.
```

In [7]:

```
f=1
```

Out[7]:

```
1
```

2. Cut the time series to the decided time interval.

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` .

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

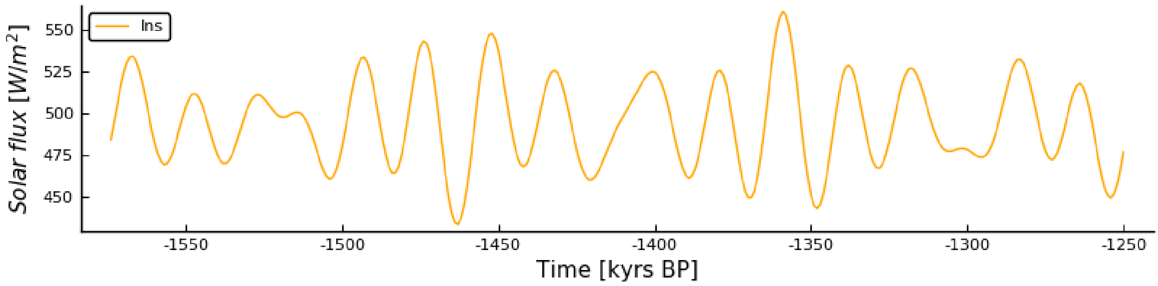
- Default version (`La2004_insol_cut`) is on a grid with 1 kyr timestep between each insolation value (as in the original dataset).

In [8]:

```
# cut out the relevant time interval (tmin:tmax) from the time array
La2004_t_cut = La2004_t_fullength[(La2004_t_fullength .>= gridstart) .& (La2004_
t_fullength .<= (gridend))]
# cut out the relevant time interval from the insolation values array
La2004_insol_cut = La2004_insol65N_fullength[(La2004_t_fullength .>= gridstart)
.& (La2004_t_fullength .<= (gridend))]

# check that we have cut the correct time interval
print(La2004_t_cut[1] : La2004_t_cut[end] , " - the cut version of La2004 time s
eries
", gridstart : gridend , " - is now the same as the common grid") # so all good


# define the time series plot

plot_La2004 =
plot(La2004_t_cut, La2004_insol_cut,
    color = :orange,
    label = "Ins",
    xlabel = "Time [kyrs BP]",
    ylabel = L"Solar \ flux \ [W/m^{2}]",
    legend = :topleft,
    grid = false,
    size = (800,200)
    )
```

```
-1574.0:1.0:-1250.0 - the cut version of La2004 time series
-1574.0:1.0:-1250.0 - is now the same as the common grid
```

Out[8]:



- High resolution (`La2004_insol_cut_hr125`)

Prepare a second version on the same grid as the Chalk pCO2 record / Grant GSL record, for high resolution analysis between the two (interpolated insolation values for every 125 years).

In [9]:

```
# La2004 hr version with time step 125 years on the common grid

# cut out the relevant time interval (tmin:tmax) from the time array
La2004_t_cut_hr125 = La2004_t_fullength_hr125[(La2004_t_fullength_hr125 .>= grid
start) .& (La2004_t_fullength_hr125 .<= gridend)]
# cut out the relevant time interval from the insolation values array
La2004_insol_cut_hr125 = La2004_insol65N_fullength_hr125[(La2004_t_fullength_hr1
25 .>= gridstart) .& (La2004_t_fullength_hr125 .<= gridend)]


# Check that the time series cut at the correct time interval (= tmin : tmax), a
nd that it has the right timestep (= hr125)
timestep = La2004_t_cut_hr125[2] - La2004_t_cut_hr125[1]
record_timegrid = La2004_t_cut_hr125[1] : timestep : La2004_t_cut_hr125[end]
begin
    if record_timegrid == commongrid_hr125
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# define the time series plot
plot_La2004_hr125 =
plot(La2004_t_cut_hr125, La2004_insol_cut_hr125,
    color = :orange,
    label =  "Ins_hr125",
    xlabel =  "Time [kyrs BP]",
    ylabel = L"Solar \ flux \ [W/m^{2}]",
    legend = :topleft,
    grid = false,
    size = (800,200)
);
```

```
UndefVarError: commongrid_hr125 not defined

Stacktrace:
 [1] top-level scope at In[9]:13
```

- High resolution (`La2004_insol_cut_hr500`)

Prepare a third version on the same grid as the Martinez-García hr record, for high resolution analysis between the two (interpolated insolation values for every 500 years).

In [10]:

```
# La2004 hr version with time step 125 years on the common grid

# cut out the relevant time interval (tmin:tmax) from the time array
La2004_t_cut_hr500 = La2004_t_fullength_hr500[(La2004_t_fullength_hr500 .>= grid
start) .& (La2004_t_fullength_hr500 .<= gridend)]
# cut out the relevant time interval from the insolation values array
La2004_insol_cut_hr500 = La2004_insol65N_fullength_hr500[(La2004_t_fullength_hr5
00 .>= gridstart) .& (La2004_t_fullength_hr500 .<= gridend)]


# Check that the time series cut at the correct time interval (= tmin : tmax), a
nd that it has the right timestep (= hr125)
timestep = La2004_t_cut_hr500[2] - La2004_t_cut_hr500[1]
record_time_grid = La2004_t_cut_hr500[1] : timestep : La2004_t_cut_hr500[end]
begin
    if record_time_grid == commongrid_hr500
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# define the time series plot
plot_La2004_hr500 =
plot(La2004_t_cut_hr500, La2004_insol_cut_hr500,
    color = :orange,
    label =  "Ins_hr500",
    xlabel = "Time [kyrs BP]",
    ylabel = L"Solar \ flux \ [W/m^{2}]",
    legend = :topleft,
    grid = false,
    size = (800,200)
    );
```

```
UndefVarError: commongrid_hr500 not defined

Stacktrace:
 [1] top-level scope at In[10]:13
```

uivD cut troubleshooting

After several attempts, we see that neither the `>=` operator nor the `=` operator work for UncertainIndexValue format that our time series are given in. It therefore cuts one value too much at the beginning and end of our grid. We will therefore set the the grid edges tmin and tmax a littlebit before and after the start and end of the common grid, so that we don't lose the first and last datapoint of the record.

In [11]:

```
#= After several attempts, we see that the >= (greater than OR EQUAL TO) operato
r does not work for uivDs (our time series format, which we want to cut).
We will therefore set the grid edges tmin and tmax a littlebit before and after
 the start and end of the common grid, so that we don't lose the datapoints on t
he edges =#

tmin = gridstart - 0.001
tmax = gridend + 0.001
;

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

LR04

In [12]:

```
?cut_timeinterval_from_uivD
```

search: **cut_timeinterval_from_uivD**

Out[12]:

No documentation found.

`cut_timeinterval_from_uivD` is a `Function`.

```
# 1 method for generic function "cut_timeinterval_from_uivD":
[1] cut_timeinterval_from_uivD(ts::UncertainIndexValueDataset, tmin,
tmax, bmp_cg::StepRange{Int64,Int64}) in Main at /Users/maria/Jottac
loud/MASTER_2.0/Koding/NBRs_ePalus_ns20/NB3_Toolbox_ns20.ipynb:In
[4]:18
```

In [13]:

```
LR04 = LR04_binned_fullength_fullageunc

# Cut out the relevant time interval of the binned LR04 time series
LR04_cut = UncertainIndexValueDataset(
    LR04.indices[(LR04.indices .> tmin) .& (LR04.indices .< tmax)], # pick out a
ll indices (ages)  where ages are larger than tmin and smaller than tmax
    LR04.values[(LR04.indices .> tmin) .& (LR04.indices .< tmax)]   # pick out a
ll values where the corresponding ages are larger than tmin and smaller than tma
x
    )
```

Out[13]:

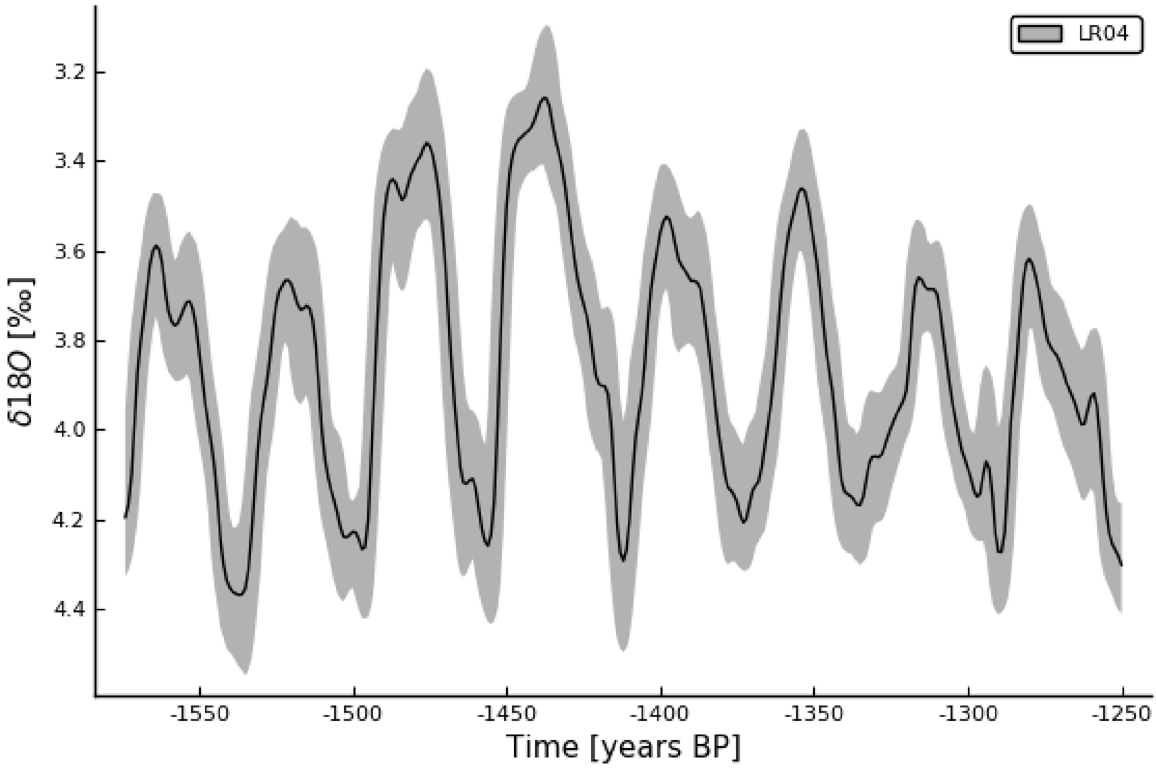
```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 325 uncertain values coupled with 325 uncertain indic
es
```

In [14]:

```
# check that the record was cut correctly and is now on the common time grid

record_timestep = LR04_cut.indices[2].value - LR04_cut.indices[1].value
record_timegrid = LR04_cut.indices[1].value : record_timestep : LR04_cut.indices
[end].value

    if record_timegrid == Binmidpoints_commongrid_1
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end


## or alternatively

binmidpoints_ts =[LR04_cut.indices[i].value for i in 1:length(LR04_cut.indices)]

    if binmidpoints_ts == Binmidpoints_commongrid_1
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end



#### 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.(LR04_cut.values, 0.5)
bin_upperq = quantile.(LR04_cut.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(LR04_cut.values, 0.025)
;

plot_LR04 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3,
    color = :black,
    label =  "LR04",
    xlabel =  "Time [years BP]",
    ylabel =  L"\delta{18}O \ [\perthousand]",
    grid = false, yflip = true
)
```

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

Out[14]:

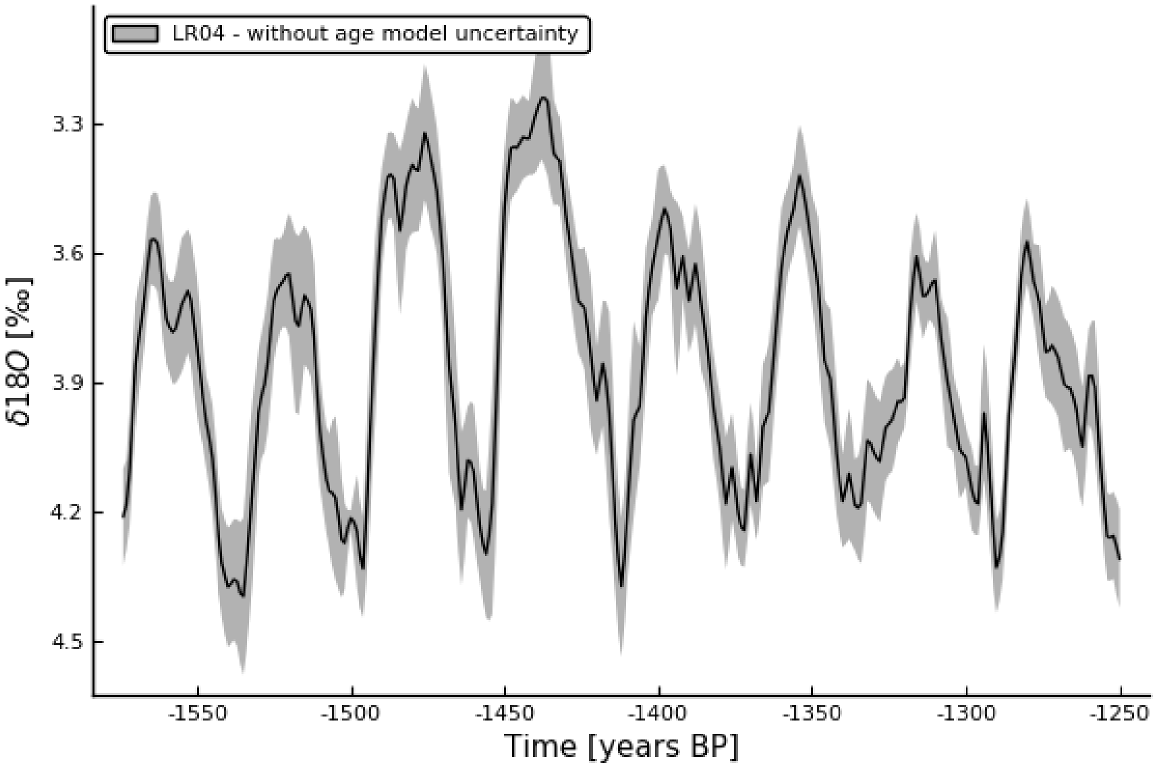


In [15]:

```
LR04 = LR04_binned_fullength_noageunc

# Cut out the relevant time interval of the binned LR04 time series
LR04_cut_noageunc = UncertainIndexValueDataset(
    LR04.indices[(LR04.indices .> tmin) .& (LR04.indices .< tmax)], # pick out a
ll indices (ages)  where ages are larger than tmin and smaller than tmax
    LR04.values[(LR04.indices .> tmin) .& (LR04.indices .< tmax)]# pick out all
 values where the corresponding ages are larger than tmin and smaller than tmax
    )


# check that the record was cut correctly and is now on the common time grid
binmidpoints_ts = [LR04_cut_noageunc.indices[i].value for i in 1:length(LR04_cut
_noageunc.indices)]
begin
    if binmidpoints_ts == Binmidpoints_commongrid_1
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end


#### 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.(LR04_cut_noageunc.values, 0.5)
bin_upperq = quantile.(LR04_cut_noageunc.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(LR04_cut_noageunc.values, 0.025)

plot_LR04_noageunc =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3, legend = :topleft,
    color = :black,
    label =  "LR04 - without age model uncertainty",
    xlabel =  "Time [years BP]",
    ylabel =  L"\delta{18}O \ [\perthousand]",
    grid = false, yflip = true
    )
```

```
The record is now on the common time grid
```

Out[15]:



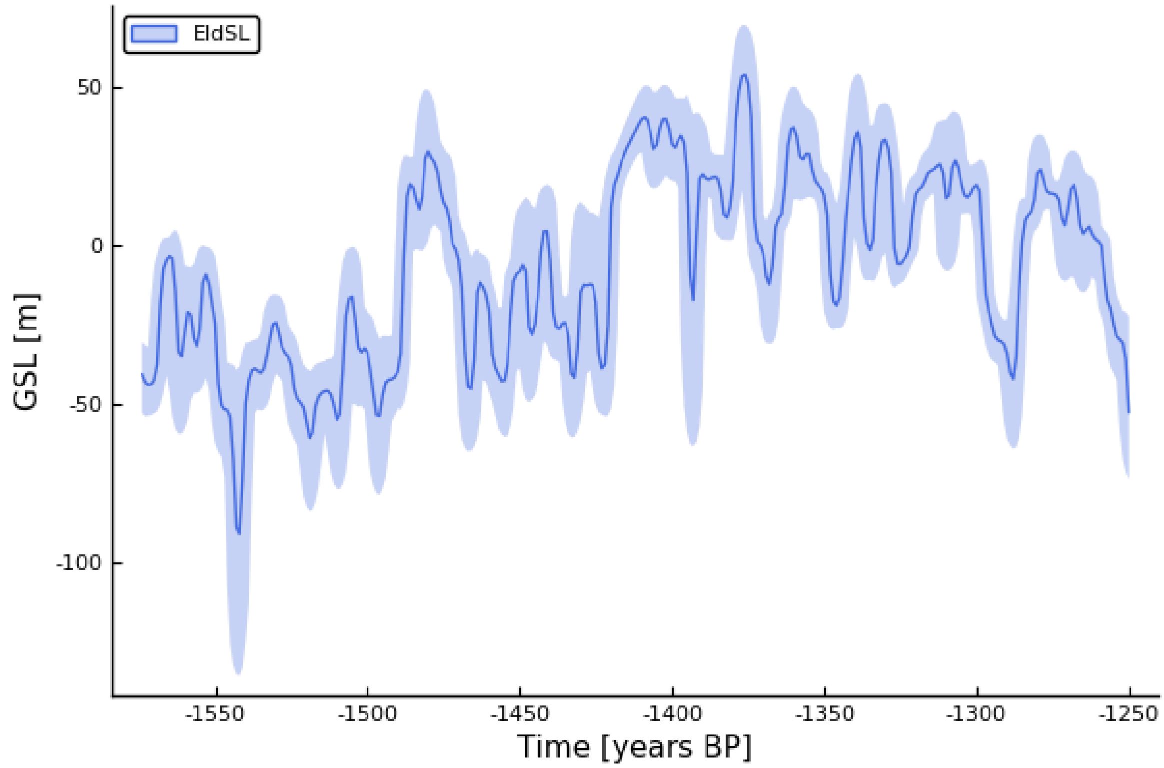
Sea level - Elderfield record

In [16]:

```
ts = E_binned_fullength_ageunc

# cut time series
ts_cut = UncertainIndexValueDataset(
    ts.indices[(ts.indices .> tmin) .& (ts.indices .< tmax)],
    ts.values[(ts.indices .> tmin) .& (ts.indices .< tmax)])


# check that the record was cut correctly and is now on the common time grid
binmidpoints_ts =[ts_cut.indices[i].value for i in 1:length(ts_cut.indices)]
begin
    if binmidpoints_ts == Binmidpoints_commongrid_1
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# save cut time series in an unambiguous name
E_cut = ts_cut

### 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 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :royalblue,
    label =  "EldSL",
    xlabel =  "Time [years BP]",
    ylabel =  "GSL [m]",
    grid = false
    )
```

The record is now on the common time grid

Out[16]:



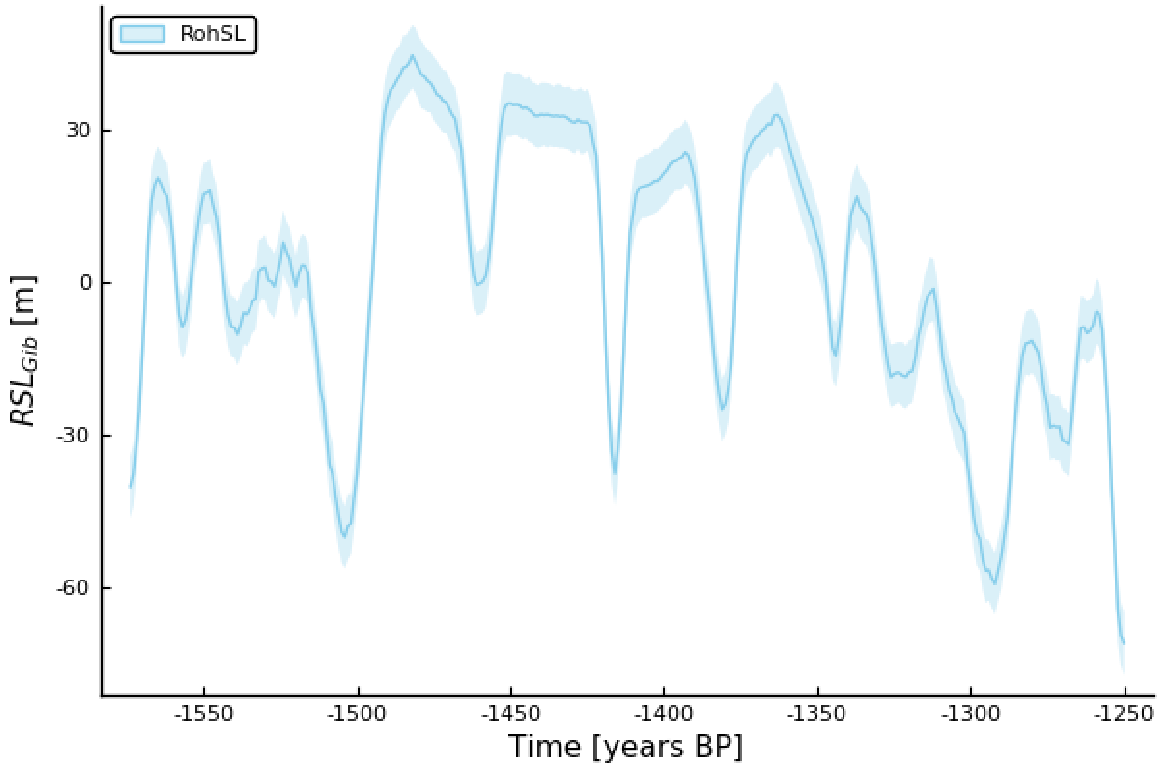
Sea level - Rohling record

In [17]:

```
ts = R_binned_full

# cut time series
ts_cut = UncertainIndexValueDataset(
    ts.indices[(ts.indices .> tmin) .& (ts.indices .< tmax)],
    ts.values[(ts.indices .> tmin) .& (ts.indices .< tmax)])

# check that the record was cut correctly and is now on the common time grid
binmidpoints_ts =[ts_cut.indices[i].value for i in 1:length(ts_cut.indices)]
begin
    if binmidpoints_ts == Binmidpoints_commongrid_1
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# save in an unambiguous name
R_cut = ts_cut


### 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)
;

binmidpoints_commongrid =[ts_cut.indices[i].value for i in 1:length(ts_cut.indic
es)]

plot_R =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :skyblue,
    label =  "RohSL",
    xlabel =  "Time [years BP]",
    ylabel =  string(L"RSL_{Gib}", " [m]"),
    grid = false
    )
```

The record is now on the common time grid

Out[17]:



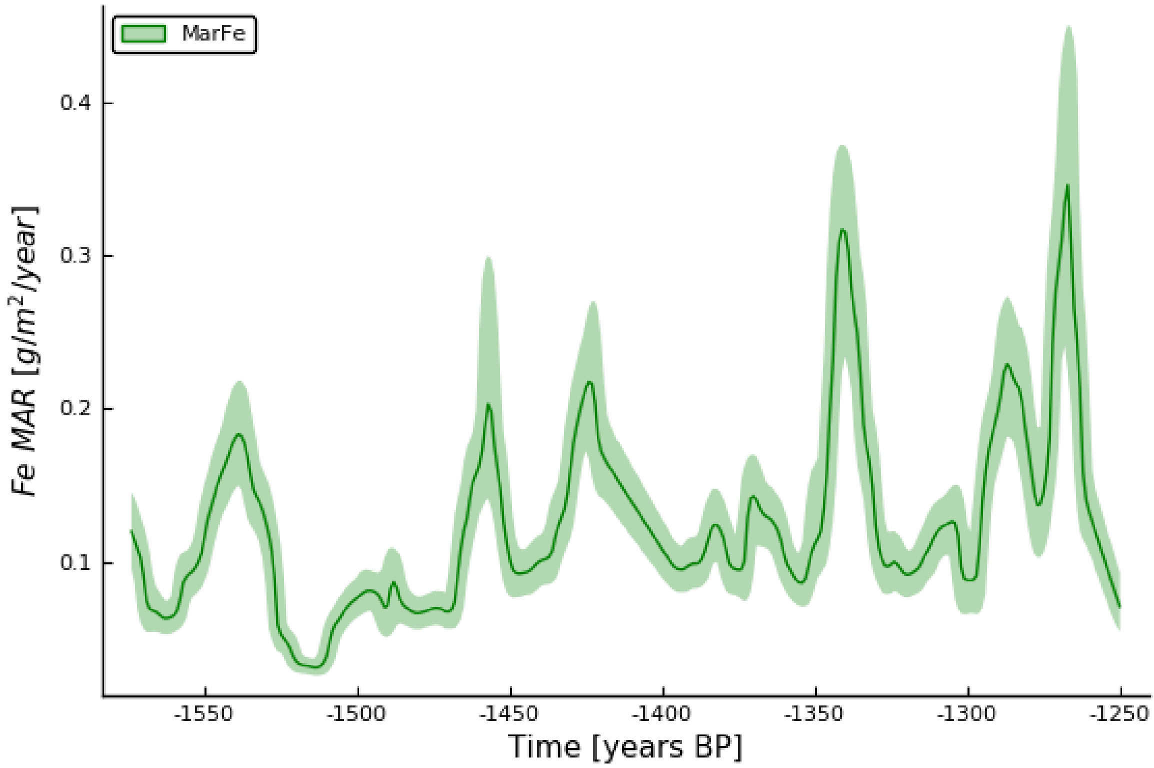
dust - Martinez-García Fe MAR record

In [18]:

```
ts = MG_binned_fullength

# cut time series
ts_cut = UncertainIndexValueDataset(
    ts.indices[(ts.indices .> tmin) .& (ts.indices .< tmax)],
    ts.values[(ts.indices .> tmin) .& (ts.indices .< tmax)])

# check that the record was cut correctly and is now on the common time grid
binmidpoints_ts =[ts_cut.indices[i].value for i in 1:length(ts_cut.indices)]
begin
    if binmidpoints_ts == Binmidpoints_commongrid_1
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# save the cut time series uivD in an unambiguous name
MG_cut = ts_cut


### 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 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :green,
    label =  "MarFe",
    xlabel =  "Time [years BP]",
    ylabel =  L"Fe \ MAR \ [g/m^{2}/year]",
    grid = false
    )
```

```
The record is now on the common time grid
```

Out[18]:



Compute the predictive asymmetry between the time series

- Compute the predictive asymmetries between each time series pair (method explained in NB2 and function defined in NB3)
- Normalize the results (function defined in NB3), and make a plot showing the 95% confidence interval

Outline for analyses

1. **$\delta^{18}O$ - insolation**

- LR04 - La2004

1. **GSL - insolation**

- Elderfield - La2004
- Rohling - La2004

1. **GSL / $\delta^{18}O$ - pCO2** (syn-MPT)
2. **GSL / $\delta^{18}O$ - dust**

- Elderfield - Martinez-García
- Rohling - Martinez-García

1. **pCO2 - dust**
2. **pCO2 - insolation**
3. **dust - insolation**

- Martinez-García - La2004

---

1. $\mathcal{A}$ between $\delta^{18}O$ and insolation

LR04 - La2004

Let's start with LR04 and insolation - because the LR04 d18O stack is tuned to the insolation signal, and this "bias" is carried on by tuning other gsl records to the LR04 stack, so it is interesting to see what to keep in mind from this.

(Both the Elderfield and the Spratt-Lisiecki sea-level records are tuned to the LR04, whose age model is based on a lag of $\delta^{18}O$ behind insolation. This is based on the understanding of the the Milankovitch cycles as a key driver behind the ice age cycles, but is however, a caveat for our method, which seeks to infer this causal connection, rather than a priori assume it. We will therefore compute the predictive asymmetry between the LR04 $\delta^{18}O$ record and the La2004 northern hemisphere summer insolation record, to ... check for bias?

> What would we expect? Drive insolation -> response $\delta^{18}O$. But we would expect this either way if it was caused by real signal or age model assumptions... Not sure what to make out as an argument here..)

Compute the predictive asymmetry between the time series pair. We compute the normalized predictive asymmetry using the function defined in NB3.

In [20]:

```
# Recall the time series on the common grid as X and Y
X = LR04_cut
Y = La2004_insol_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/LR04_La2004.
jld2")
```

```
Results are saved in the .jld2 file.  26.300301 seconds (140.88 M al
locations: 10.264 GiB, 19.09% gc time)
```

In [19]:

```
# load the results to see the arrays saved in hte computation
@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/LR04_La2004.jld2"
```

Out[19]:

```
7-element Array{Symbol,1}:
 :X
 :Y
 :ηmax
 :rsTE_XtoY
 :rsPA_XtoY
 :rsTE_YtoX
 :rsPA_YtoX
```

In [20]:

```
# normalize the results for comparability

@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/LR04_La2004.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 confidence interval we want to plot (2σ, aka 95%).

In [21]:

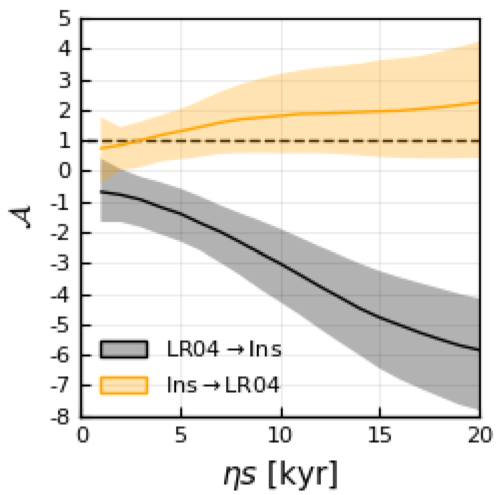
```
### Calculate the quantiles for the confidence interval we want to plot (2σ, aka 95%)

    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax] # median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normPA_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax] # median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normPA_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for i in 1:ηmax] # lower quantile
;
```

Define the results plot

In [22]:

```
# define the predictive asymmetry plot
plot_normPA_LR04_La2004 =
plot(#title = string("Normalized # define the predictive asymmetry plot", L"\mat
hcal{A}", ") between ", L"\delta^{18}O", " (LR04) and northern hemisphere summer
insolation (La2004)"),
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-8,5), yticks = (-10:1:10),
    legend = :bottomleft, bg_legend = :transparent,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}",
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label =  "" # mean TE
    )
plot!(normPA_XtoY_median, # ...from X to Y
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("LR04", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :black
    )
plot!(normPA_YtoX_median, # ...from Y to X
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "LR04"),
    fillalpha = 0.3,  color = :orange
    )
```

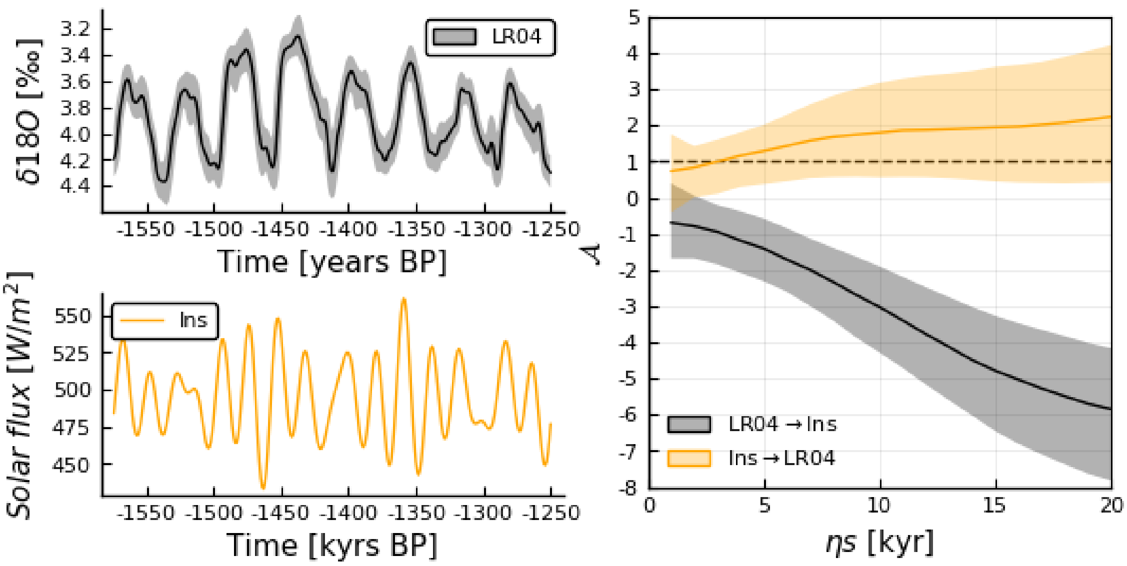
Out[22]:



Plot the figure to include in thesis, including the time series and their $\mathcal{A}$ results as subplots in one figure.

In [23]:

```
# Plot overview of time series pair to analyse
plot_overview_LR04_La2004 =
plot(layout = grid(2,1),
    size = (500, 400),
    plot_LR04,
    plot_La2004
    )

# join plots of time series and pa results to a results subplot
plot_results_LR04_La2004 =
plot(size = (600,300),
    layout = grid(1,2),
    plot_overview_LR04_La2004,
    plot_normPA_LR04_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/LR04_La2004.pdf")
# This is the version with age model uncertainty
```



Caption: *Normalized predictive asymmetry between $\delta^{18}O_b$ and northern hemisphere summer insolation.* Insolation (orange) is the La2004 numerical solution of insolation intensity on June 31^{st} at $65^{\circ}N$ (Laskar et al., 2004).$\delta^{18}O_b$, shown in black, is the LR04 global stack by Lisiecki & Raymo (2005). Note that the age model of this record is built on the premiss that the \d18O signal *follows* northern hemisphere insolation intensity with an assumed lag of 300 years. This poses a caveat in the results of our analysis.

- Without age model uncertainty

In [156]:

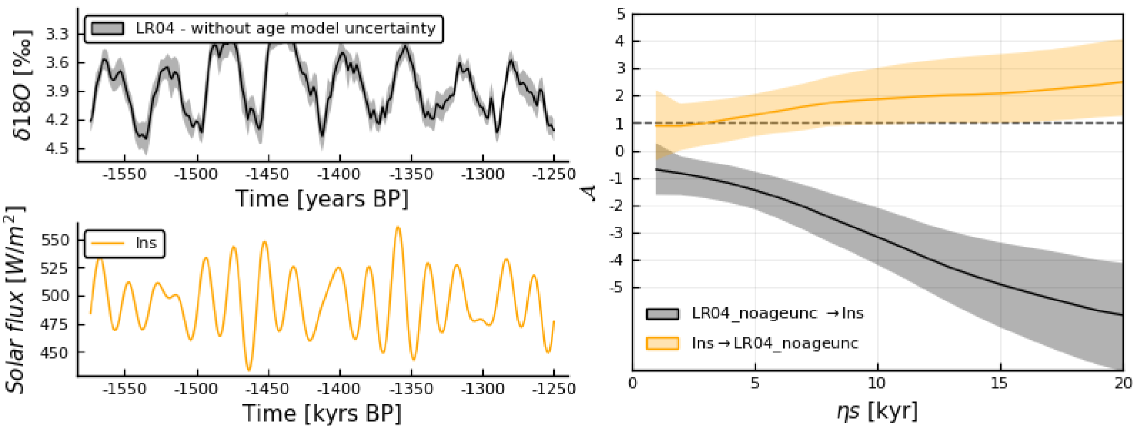
```
# Recall the time series on the common grid as X and Y
X = LR04_cut_noageunc # this is the version without age model uncertainty  inclu
ded
Y = La2004_insol_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/LR04noageunc
_La2004.jld2")
```

```
Results are saved in the .jld2 file.  26.720721 seconds (141.99 M al
locations: 10.350 GiB, 18.53% gc time)
```

In [24]:

```
# Load and normalize the results

@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/LR04noageunc_La2004.j
ld2"
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

    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;

# defining the results plot
plot_normPA_LR04_La2004_noageunc =
plot(xlims = (0, ηmax),xticks = (0 : 5 : ηmax),
    ylims = (-8,5), yticks = (-5:1:5),
    legend = :bottomleft, bg_legend = :transparent,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label =  "" # mean TE
    )
plot!(normPA_XtoY_median, # ...from X to Y
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("LR04_noageunc ", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :black
    )
plot!(normPA_YtoX_median, # ...from Y to X
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "LR04_noageunc "),
    fillalpha = 0.3, color = :orange
    )

# defining the overview plot
plot_overview_LR04_La2004_noageunc =
plot(plot_LR04_noageunc, plot_La2004, layout = (2,1))


# join plots of time series and pa results to a results subplot
plot_results_LR04_La2004_noageunc =
plot(size =  (800,300),
```

```
    layout = grid(1,2),
    plot_overview_LR04_La2004_noageunc,
    plot_normPA_LR04_La2004_noageunc)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/LR04noageunc_La200
4.pdf") # this is the version without age model uncertainty included
```



2. $\mathcal{A}$ between sea level and insolation

Elderfield - La2004

In [49]:

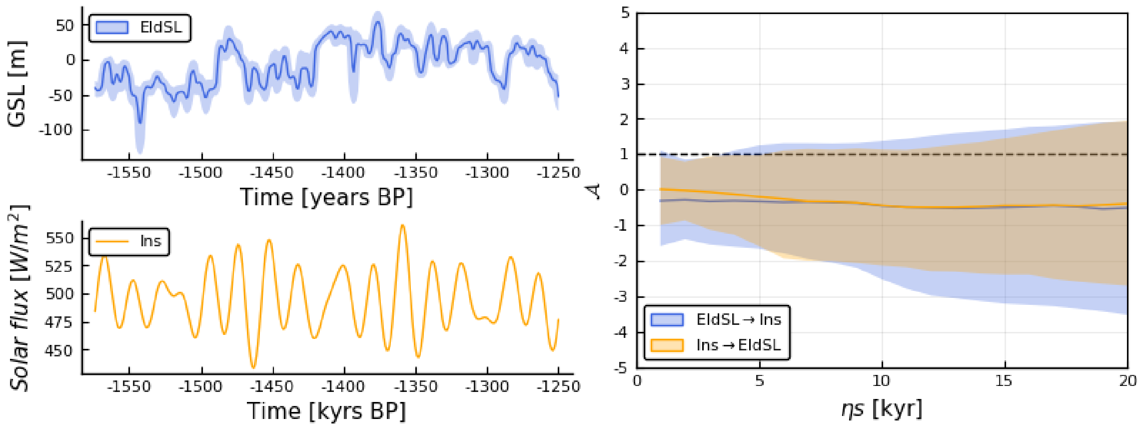
```
# Compute the predictive asymmetry between the time series

# First, recall the time series on the common grid as X and Y
X = E_cut
Y = La2004_insol_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(
    X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/E_La2004.jld
2"
    )
```

```
Results are saved in the .jld2 file.  31.137012 seconds (141.66 M al
locations: 10.326 GiB, 17.66% gc time)
```

In [25]:

```
# Load and normalize the predictive asymmetry results

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/E_La2004.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 to plot the 95% confidence interval
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_E_La2004 =
plot(xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5),
    yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label = "" # mean TE
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("EldSL", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :royalblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "EldSL"),
    fillalpha = 0.3,  color = :orange
    )
;


# join plots of time series and pa results to a results subplot
plot_overview_E_La2004 =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_E,
```

```
    plot_La2004
    )

plot_results_E_La2004 =
plot(size =  (800,300),
    layout = grid(1,2),
    plot_overview_E_La2004,
    plot_normPA_E_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/E_La2004.pdf")
```



Rohling - La2004

In [51]:

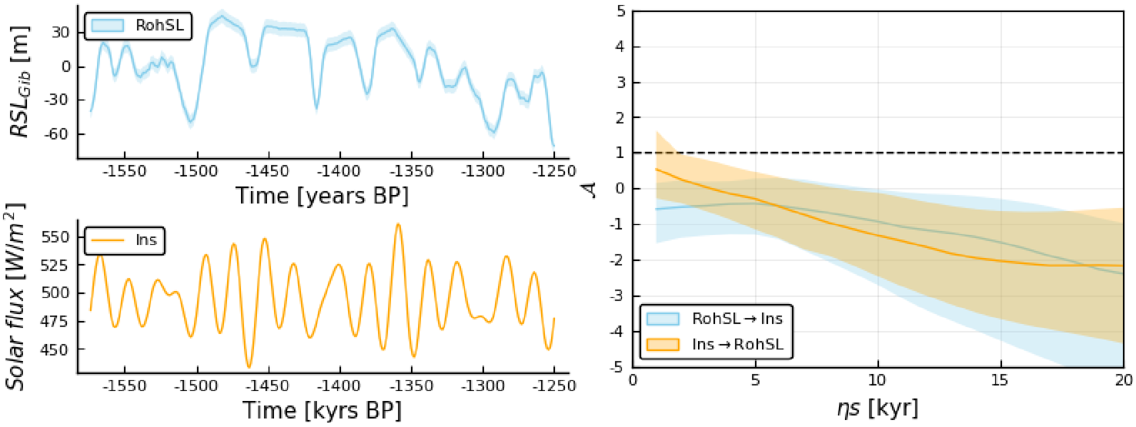
```
# Recall the time series on the common grid as X and Y
X = R_cut
Y = La2004_insol_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/R_La2004.jld
2")
```

```
Results are saved in the .jld2 file.  25.615964 seconds (141.96 M al
locations: 10.348 GiB, 18.69% gc time)
```

In [26]:

```
# Load and normalize the predictive asymmetry results

# Load and normalize the results
@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/R_La2004.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 to plot the median and the 95% confidence interval
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_R_La2004 =
plot(#title =
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label =  "" # mean TE
    )
plot!(normPA_XtoY_median, # ...from X to Y
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("RohSL", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :skyblue
    )
plot!(normPA_YtoX_median, # ...from Y to X
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "RohSL"),
    fillalpha = 0.3, color = :orange)
;


# join plots of time series and pa results to a results subplot
plot_overview_R_La2004 =
plot(layout = grid(2,1),
    size = (800, 400),
    plot_R,
```

```
    plot_La2004
    )

plot_results_R_La2004 =
plot(size =  (800,300),
    layout = grid(1,2),
    plot_overview_R_La2004,
    plot_normPA_R_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/R_La2004.pdf")
```



4. $\mathcal{A}$ between sea level / $\delta^{18}O$ and dust

LR04 - Martinez-García

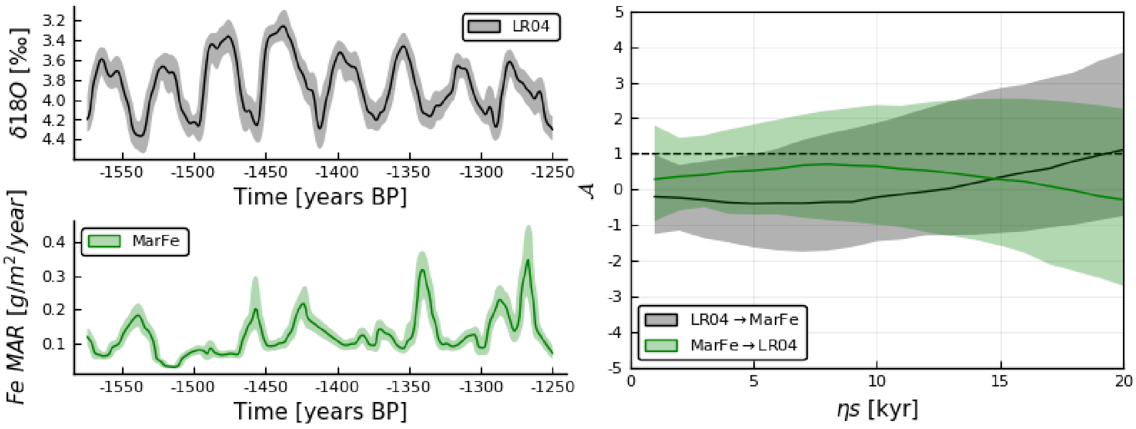
In [ ]:

```
#Compute the predictive asymmetry between the two time series

# recall the time series on the common grid as X and Y
X = LR04_cut
Y = MG_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/LR04_MG.jld
2")
```

In [27]:

```
# Load and normalize the predictive asymmetry results

@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/LR04_MG.jld2"
# unpack from structure
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
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_LR04_MG =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("LR04", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :black)
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "LR04"),
    fillalpha = 0.3,  color = :green)
;


# join plots of time series and pa results to a results subplot
plot_overview_LR04_MG =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_LR04,
    plot_MG
```

```
    )

plot_results_LR04_MG =
plot(size =  (800,300),
    layout = grid(1,2),
    plot_overview_LR04_MG,
    plot_normPA_LR04_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/LR04_MG.pdf")
```



Elderfield - Martinez-García

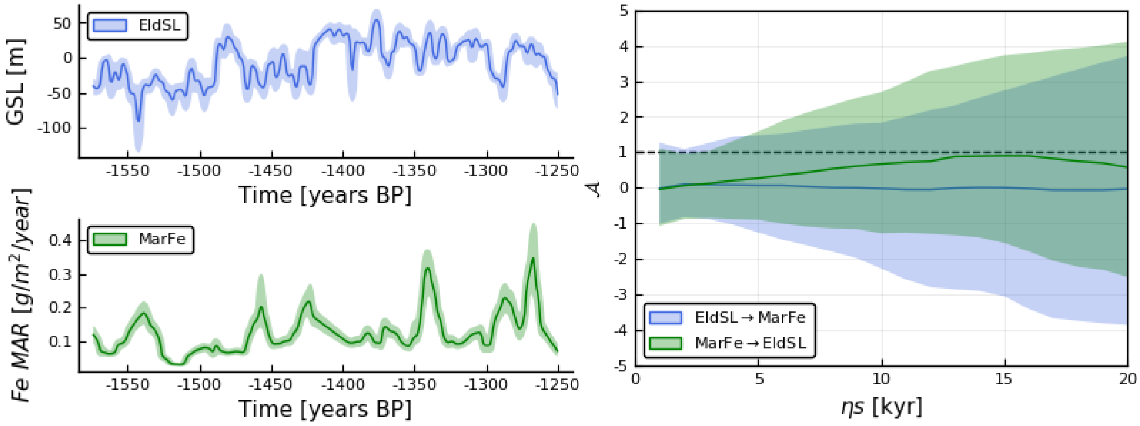
In [ ]:

```
# recall the time series on the common grid as X and Y
X = E_cut
Y = MG_cut

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/E_MG.jld2")
```

In [28]:

```
# Plot the results, showing the 95% confidence interval

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/E_MG.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
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_E_MG =
plot(#title = L"$\mathcal{A}$ between GSL and Southern Ocean Fe deposition",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("EldSL", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :royalblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, color = :green)
;


# join plots of time series and pa results to a results subplot
plot_overview_E_MG =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_E,
```

```
    plot_MG
    )

plot_results_E_MG =
plot(size =  (800,300),
    layout = grid(1,2),
    plot_overview_E_MG,
    plot_normPA_E_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/E_MG.pdf")
```



Rohling - Martinez-García

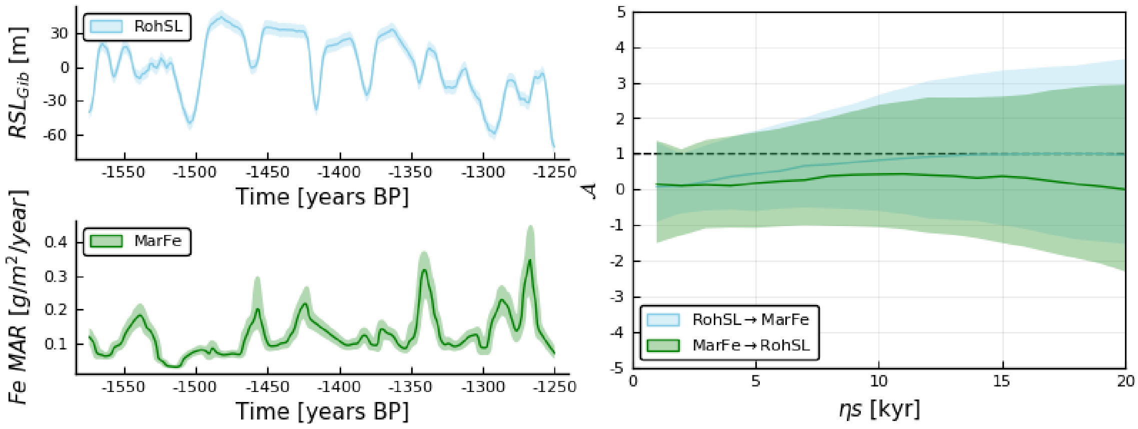
In [ ]:

```
#Compute the predictive asymmetry between the two time series

# recall the time series on the common grid as X and Y
X = R_cut
Y = MG_cut

# Compute the predictive asymmetry (function defined in NB3)
computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/R_MG.jld2")
```

In [29]:

```
# Plot the results, showing the 95% confidence interval

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/R_MG.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
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_R_MG =
plot(xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("RohSL", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :skyblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "RohSL"),
    fillalpha = 0.3, color = :green)
;


# join plots of time series and pa results to a results subplot
plot_overview_R_MG =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_R,
    plot_MG
```

```
    )

plot_results_R_MG =
plot(size =  (800,300),
    layout = grid(1,2),
    plot_overview_R_MG,
    plot_normPA_R_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/R_MG.pdf")
```



7. $\mathcal{A}$ between insolation and dust

La2004 - Martinez-García

In [56]:

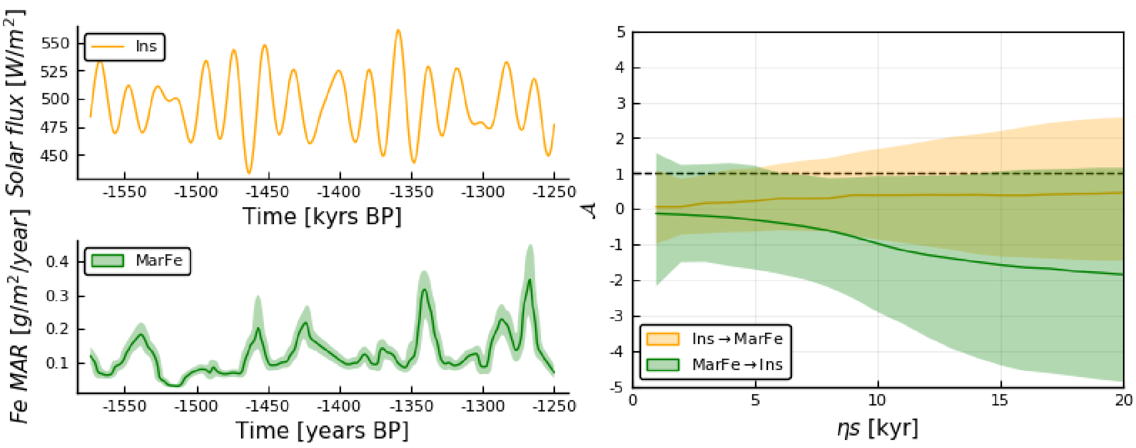
```
# recall the time series on the common grid as X and Y
X = La2004_insol_cut
Y = MG_cut

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/La2004_MG.jl
d2")
```

```
Results are saved in the .jld2 file.  25.692350 seconds (141.13 M al
locations: 10.286 GiB, 18.98% gc time)
```

In [30]:

```
# Plot the results, showing the 95% confidence interval

# load and normalize the predictive asymmetry results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/preMPT_1500/La2004_MG.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
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_La2004_MG =
plot(#title = L"$\mathcal{A}$ between insolation and Fe dust"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250),
    grid = true, border = true,
    hline([1], line = (:dash, :black)),
    label =  "")
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("Ins", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :orange)
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :green)


# join plots of time series and pa results to a results subplot
plot_overview_La2004_MG =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_La2004,
    plot_MG)

plot_results_La2004_MG =
```

```
plot(size =  (800,300),
    layout = grid(1,2),
    plot_overview_La2004_MG,
    plot_normPA_La2004_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/preMPT_1500/La2004_MG.pdf")
```



- high resolution analysis with time step of 500 years?

> We decided not to interpolate the Maritnez-Garcia record for this, as the 500 yr time step is higher resolution than the mean resolution across the time interval (900 years)

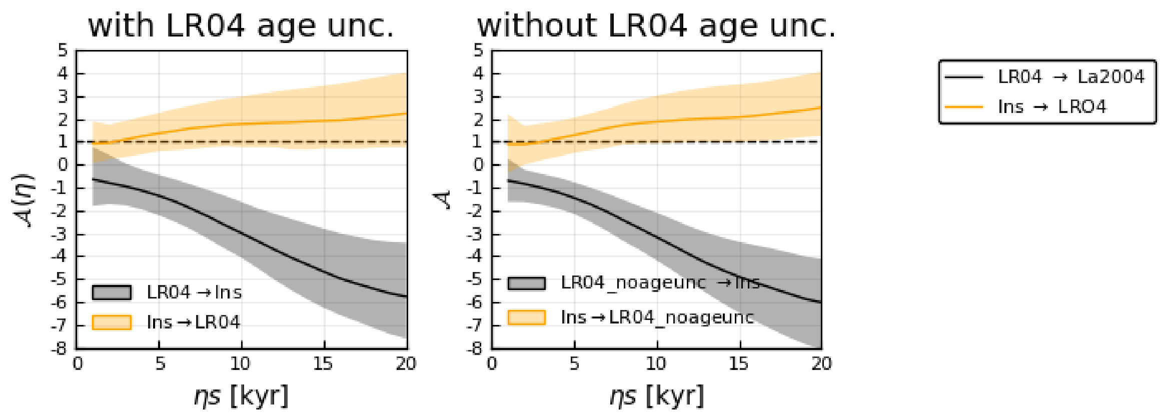
Ensemble plots of results

1. $\delta^{18}O$ **- insolation**

In [212]:

```
p01 = plot(plot_normPA_LR04_La2004, legend = :bottomleft, bg_legend = :transparent, title = "with LR04 age unc.")
p02 = plot(plot_normPA_LR04_La2004_noageunc, legend = :bottomleft, bg_legend = :transparent, title = "without LR04 age unc.")

legend = plot( # empty plot for legend
    [0 0],
    #ribbon = [(0,0) (0,0)], # How to make "fat" labels?
    showaxis = false, grid = false, legend = :outertopright,
    label = [string("LR04 ", L"\rightarrow", " La2004") string("Ins ", L"\rightarrow", " LRO4") ],
    color = [:black :orange])

plot(p01, p02, legend,
    layout = @layout([[A B ] E{0.1w}]),size = (700,250),
    ylims = (-8,5), yticks = (-10:1:10),
    border = :true
    )
```

Out[212]:

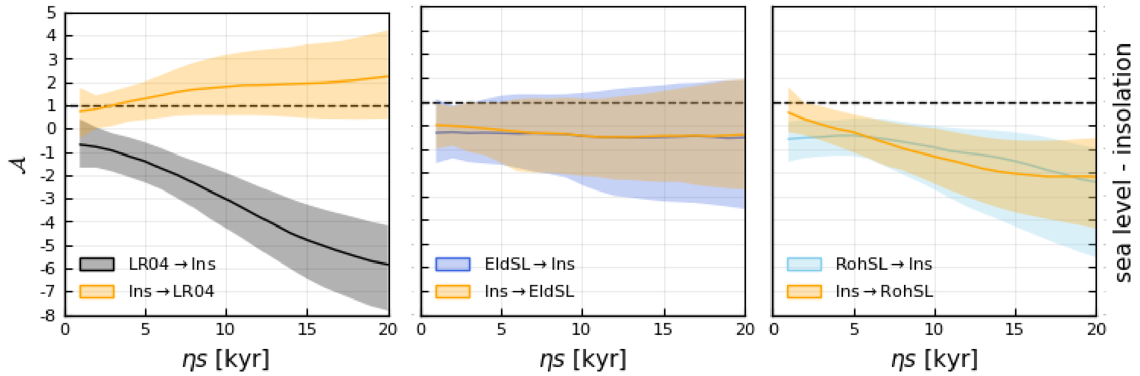


Not a very big difference actually...

1. **GSL - insolation**

In [31]:

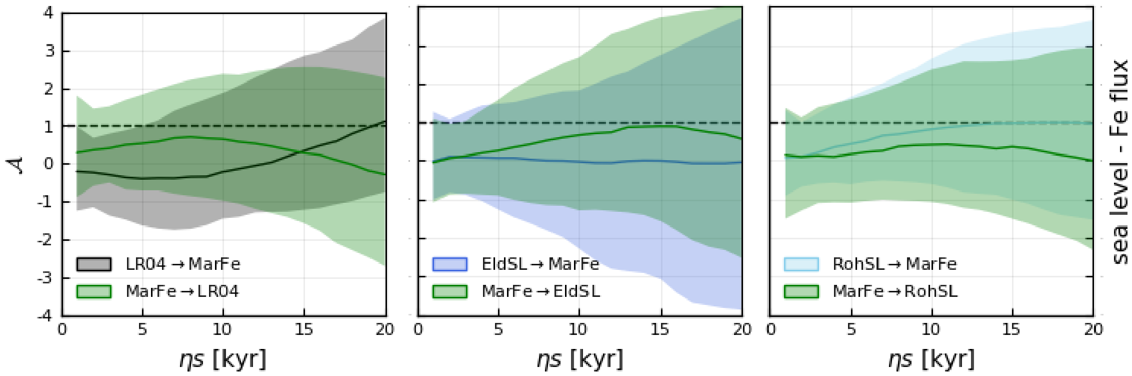
```
# remove the legends from the individual plots

p11 = plot(plot_normPA_LR04_La2004, legend = :bottomleft, bg_legend = :transpare
nt, title = "", ylabel = L"\mathcal{A}")
p12 = plot(plot_normPA_E_La2004,    legend = :bottomleft, bg_legend = :transpare
nt, ylabel = "", ytickfont = false)
p13 = plot(plot_normPA_R_La2004,    legend = :bottomleft, bg_legend = :transpare
nt, ylabel = string("sea level - insolation"), ymirror = true, ytickfont = false
)


# plot the 6 results plots and the legend plot to the right
pe_gsl_insol = plot(
    p11, p12, p13,
    layout = grid(1,3), size = (750,250),
    border = true, xaxis = :on,
    ylims = (-8,5), yticks = (-10:1:10),
    )

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/preMPT_1500/gsl_insol.p
df")
```



1. **GSL - dust**

In [33]:

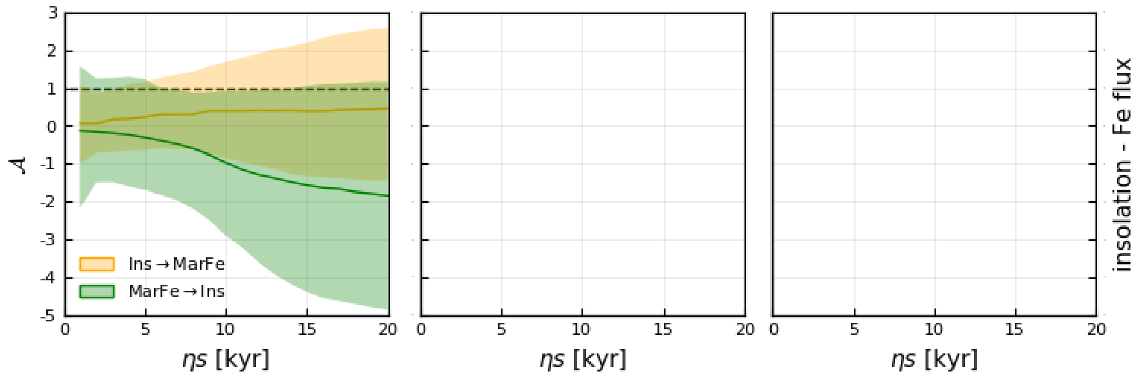
```
p41 = plot(plot_normPA_LR04_MG, ylabel = L"\mathcal{A}")
p42 = plot(plot_normPA_E_MG, ylabel = "", ytickfont = false)
p43 = plot(plot_normPA_R_MG, ylabel = string("sea level - Fe flux"), ymirror = t
rue, ytickfont = false)


pe_gsl_dust = plot(
    p41, p42, p43,
    layout = grid(1,3), size = (750,250),
    border = true, xaxis = :on,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-4,4), yticks = (-10:1:10),
    )

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/preMPT_1500/gsl-dust.pd
f")
```



1. **insolation - dust**

In [34]:

```
p74 = plot(plot_normPA_La2004_MG, ylabel =  L"\mathcal{A}")
#p75 = plot(plot_normPA_La2004_MG_hr500, ylabel = "", ytickfont = false)
p_empty1 = plot(size = (250,250), border = false, xlabel =  string(L"ηs"," [ky
r]"), xticks = (0:5:20),lims = (0,20), yticks = :none, ytickfont = false)
p_empty2 = plot(size = (250,250), border = false, xlabel =  string(L"ηs"," [ky
r]"), xticks = (0:5:20),lims = (0,20), yticks = :none, ytickfont = false, ylabel
= string("insolation - Fe flux"), ymirror = true)


pe_insol_dust = plot(
    #plot_normPA_La2004_MG, size = (250,250),
    p74, p_empty1, p_empty2, layout = grid(1,3), size = (750,250),
    border = true,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-5,3), yticks = (-10:1:10),
    )

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/preMPT_1500/e_insol_dus
t_preMPT_1500.pdf")
```



Overview: predictive asymmetry for the `preMPT_1500` grid

In [37]:

```
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"]
;
length(panel_labels)
```

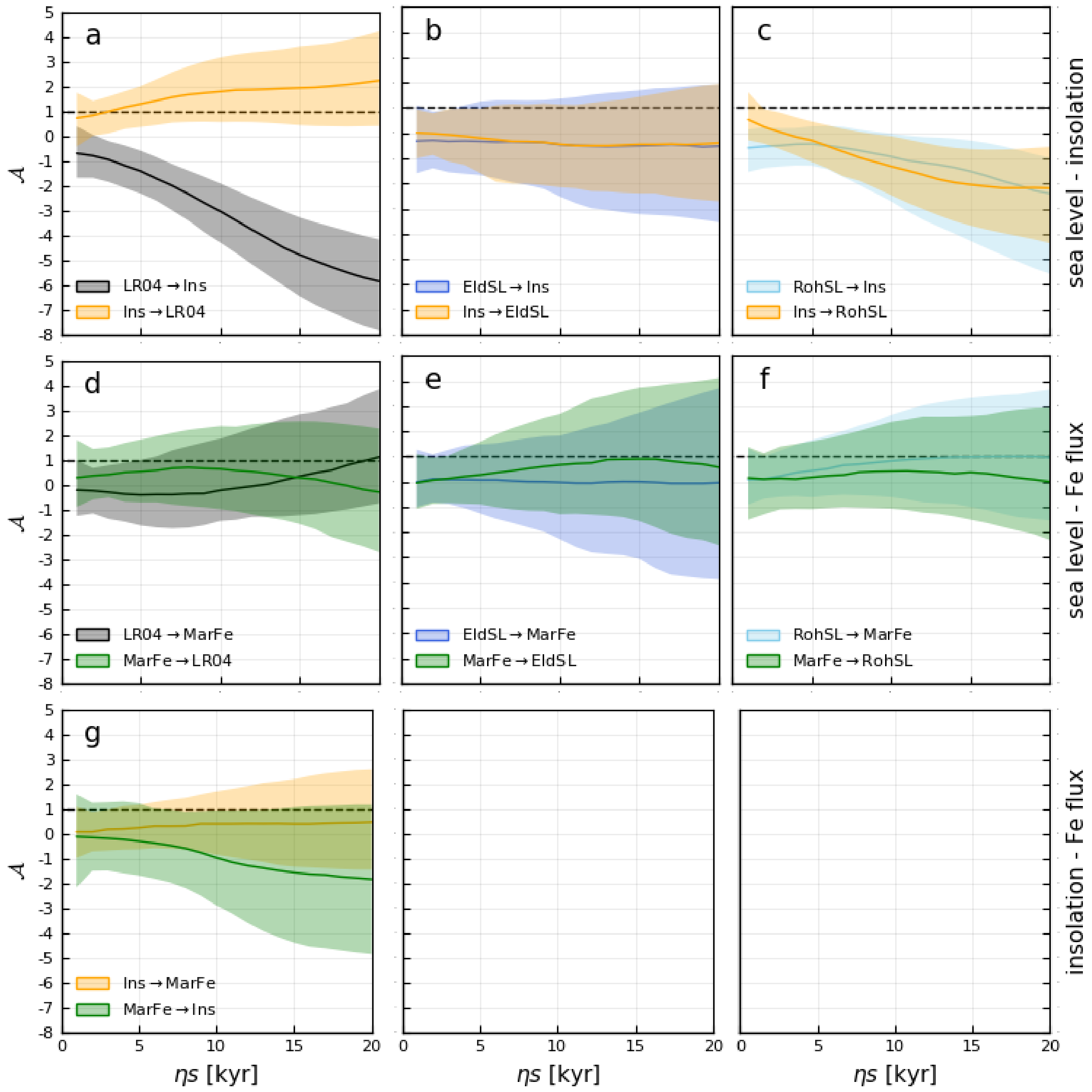
Out[37]:

```
26
```

Make the overview ensemble plot of all the predictive asymmetry results gathered in this notebook

In [39]:

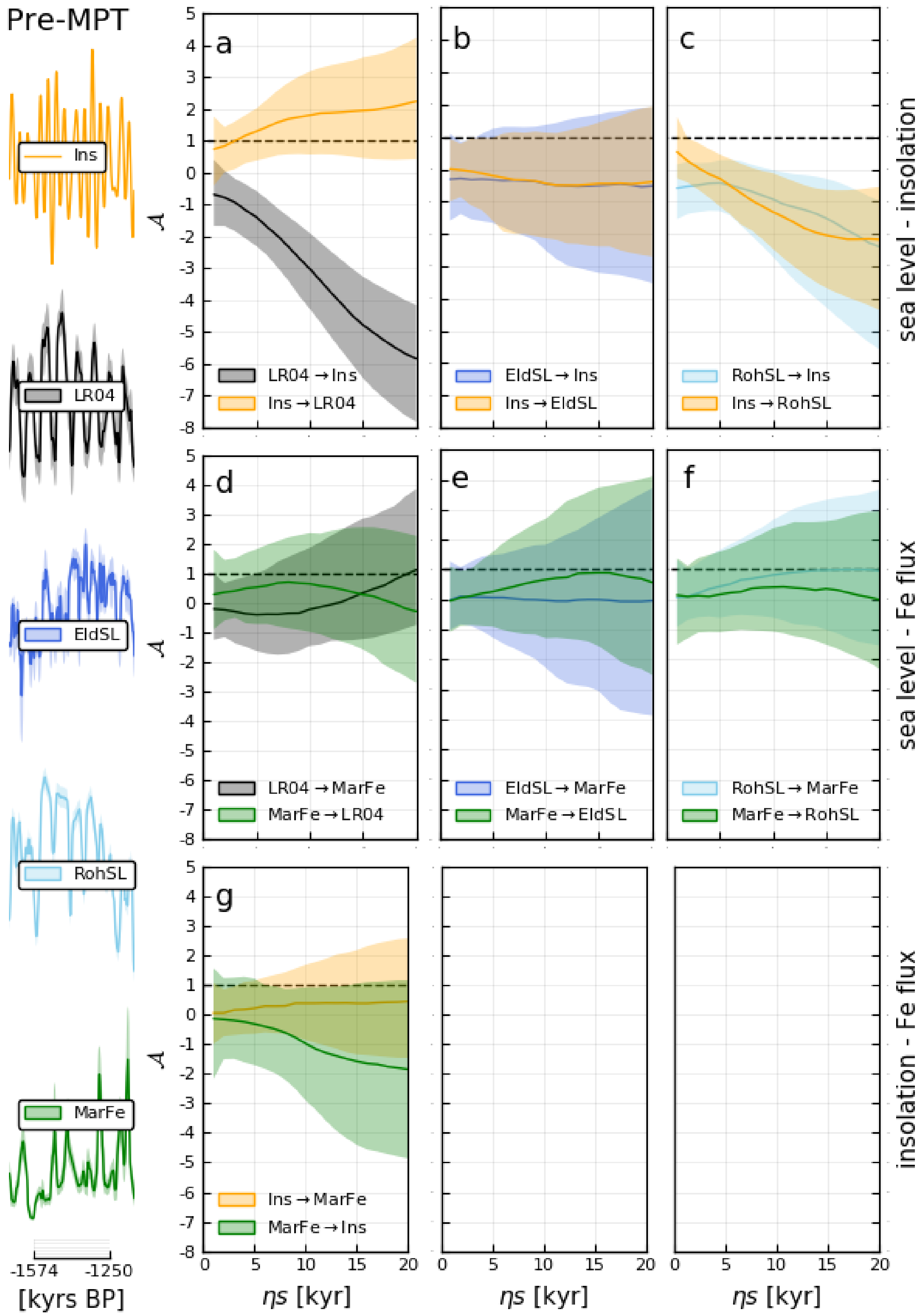
```
p101 = plot(pe_gsl_insol, xlabel = "", xtickfont = false)
    #pe_gsl_co2,
p103 = plot(pe_gsl_dust, xlabel = "", xtickfont = false)
    #pe_fit3,
    #pe_co2_dust,
    #pe_insol_co2,
p105 = plot(pe_insol_dust)


pe_allresults =
plot(p101,p103, p105,
    layout = grid(3,1), size = (750, 3*250),
    ylims = (-8,5)
    )

for i in 1:7
    annotate!(2,4, panel_labels[i], subplot = i)
end

pe_allresults
savefig("../../results_ePalus_ns20/e_allresults/MA__preMPT_1500_wPanellabels.pd
f")
```



Create master figure for appendix

In [40]:

```
# overview time series

po0 = plot(plot_La2004, xlabel = "", xaxis = :off, title = "Pre-MPT ")
#    (1240-1090 ka BP)")
po1 = plot(plot_LR04, xlabel = "", xaxis = :off)
po3 = plot(plot_E, xlabel = "", xaxis = :off)
po4 = plot(plot_R, xlabel = "", xaxis = :off)
po6 = plot(plot_C, xlabel = "", xaxis = :off)
po8 = plot(plot_MG, xlabel = "", xaxis = :off)
p_xaxis = plot(xlabel = "[kyrs BP]", xaxis = :on, xmirror = false)

l = @layout [a;b;c;d;e;f{0.01h}]

po_alltimeseries = plot(po0,po1,po3,po4,po8, p_xaxis,
    layout = l, #grid(7,1),
    size = (250, 600),
    xlims = (gridstart, gridend), xticks = (gridstart : gridend-gridstart  : gri
dend),
    ylabel = "", ytickfont = false, yaxis = :off,
    legend = :left, bg_legend = :white)

#savefig("../../results_ePalus_ns20/timeseries/po_all_synMPT_Cgrid.pdf")


l = @layout [a{0.1w} b{0.9w}]

plot(po_alltimeseries, pe_allresults, layout = l, size = (72*8.27, 72*11.79)) #
 A4 size
savefig("../../results_ePalus_ns20/e_allresults/MA__preMPT_1500_Appendix.pdf")
```


Pre-MPT
Ins
LR04
EldSL
RohSL
MarFe
-1574
-1250
[kyrs BP]
a
b
c
d
e
f
g
LR04 → Ins
Ins → LR04
EldSL → Ins
Ins → EldSL
RohSL → Ins
Ins → RohSL
LR04 → MarFe
MarFe → LR04
EldSL → MarFe
MarFe → EldSL
RohSL → MarFe
MarFe → RohSL
Ins → MarFe
MarFe → Ins
sea level - insolation
sea level - Fe flux
insolation - Fe flux
ηs [kyr]

In [ ]:

```

```