time grid label for this notebook: `NBR_full_800`

Predictive asymmetry analysis for Late Pleistocene Earth system dynamics.

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 covering the last 480 kyrs leading up to the Holocene. 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 after 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"
```

```
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.  37.6182
21 seconds (164.25 M allocations: 11.102 GiB, 13.72% gc time)
```

The following records are used for analysis over this notebook's time interval (record labels in bold / code labels in italics):

- **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)*.
- **SpraSL** / SL: Global sea level stack (principal component analysis) of 7 sea level records, whereof 4 spanning the last 800 kyrs, by *Spratt & Lisiecki (2016)*.
- **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)*.
- **BerCO2** / B: Record of atmospheric pCO2 spanning the last 800 kyrs, from Epica Dome C ice core, East Antarctica, by *Bereiter et al. (2015)*.
- **ChaCO2** / C: High resolution $\delta^{11}B$ proxy record for pCO2, spanning a 150 kyr interval of the MPT, by *Chalk et al. (2017)*.
- **IceDust** / L: Record of Antarctic dust accumulation from Epica Dome C ice core, spanning the past 800 kyrs, by *Lambert et al. (2008)*.
- **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. **Set the time series to a common time grid**

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

1. **Run pairwise analyses for predictive asymmetry between the time series**

- Compute the normalized predictive asymmetry (function defined in NB3) for each time series pair, and save the results in a .jld2 file.
- Produce results plot of the normalized predictive asymmetry for each time series pair.

1. **Produce overview plots of predictive asymmetry results**

- Produce the 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.

---

1. Set the time series to 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.

- 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


#### We select a time interval according to the periods covered by the records

# the record with the "latest start" defines the start of the common grid
gridstart = maximum([
        tmin_LR04,
        tmin_La2004,
        tmin_SL,
        tmin_E,
        #tmin_G, # the Grant record is the shortest record and delimits the time
interval analysed.
        #tmin_R,
        tmin_B,
        # tmin_C, # the Chalk record is left out as it does not cover the post-M
PT time interval
        tmin_L,
        tmin_MG
        ])

# the record with the "earliest end" the end of the common grid
gridend = minimum([
        tmax_LR04,
        tmax_La2004,
        tmax_SL,
        tmax_E,
        #tmax_G,
        tmax_R,
        tmax_B,
        # tmax_C,
        tmax_L,
        tmax_MG
        ])

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 797.0 ka BP to 1
3.0 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 for the common grid
binsize_1 = 1          # 1 kyrs - is the default timestep on which all time seri
es are binned (NB1)

# Additionally, we had prepared some records for higher resolution analyses
binsize_hr125 = 0.125   # High resolution (hr) records are additionally binned o
n a hr grid, 125 year timestep (Grant, Chalk, La2004)
binsize_hr500 = 0.5  # Martinez-García and La2004 are additionally binned on a g
rid with a 500 year timestep

# (but the main timestep for all analyses will be 1000 years)
print("All records are on a regular grid with timestep of ", binsize_1, " kyr.
Additional high resolution analyses have timesteps of ", binsize_hr125 , " and "
,binsize_hr500, " kyrs.")
```

```
All records are on a regular grid with timestep of 1 kyr.
Additional high resolution analyses have timesteps of 0.125 and 0.5
kyrs.
```

We can now define the time grid for analyses in this notebook. Objects associated with this grid are labeled with the grid suffix `full_800` .

In [4]:

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

commongrid = gridstart : binsize_1 : gridend
Binmidpoints_commongrid =[commongrid[i] for i in 1:length(commongrid)]

# Additional high resolution grids
commongrid_hr125 = gridstart : binsize_hr125 : gridend
Binmidpoints_commongrid_hr125 = [commongrid_hr125[i] for i in 1:length(commongri
d_hr125)]

commongrid_hr500 = gridstart : binsize_hr500 : gridend
Binmidpoints_commongrid_hr500 = [commongrid_hr500[i] for i in 1:length(commongri
d_hr500)]

### But this is the main grid
print(gridstart : binsize_1 : gridend , " defines the main common grid for analy
ses in this 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.) =#
Binmidpoints_commongrid_hr500;
```

```
-797.0:1.0:-13.0 defines the main common grid for analyses in this n
otebook.
```

Binning of state space (ϵ) for transfer entropy estimation in the following analyses

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 the bins we use to count visitations), which is given by ϵ. We choose ϵ according to the length of our time series, as given by the Palus horizon. This is all defined inside our `function_from_XY_to_normPA` , see NB3, and NB2 for explanation for further information. **For overview, let's check the ϵ used for the time series in this notebook**.

In [5]:

```
# Palus horizon

# Palus horizon
N = length(Binmidpoints_commongrid)
eD = 3
ϵ = Int(round( N^(1/(eD+1)) ))

print("binning resolution ϵ used to estimate transfer entropy is
    ", ϵ, " for the 1 kyr grid (time series length N = ", N,")."
    )

# Palus horizon

# Palus horizon
N = length(Binmidpoints_commongrid_hr500)
eD = 3
ϵ = Int(round( N^(1/(eD+1)) ))

print("
    ", ϵ, " for the 500 yr grid (time series length N = ", N,")."
    )


N = length(Binmidpoints_commongrid_hr125)
eD = 3
ϵ = Int(round( N^(1/(eD+1)) ))

print("
and ", ϵ, " for the 125 yr grid, (time series length N = ", N,")."
    )
```

```
binning resolution ϵ used to estimate transfer entropy is
    5 for the 1 kyr grid (time series length N = 785).
    6 for the 500 yr grid (time series length N = 1569).
and 9 for the 125 yr grid, (time series length N = 6273).
```

---

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 (AnalySeries computation)

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

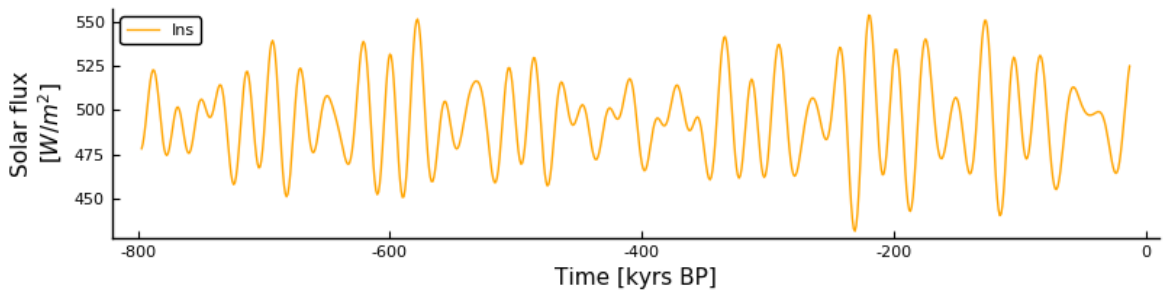
- Default version (`La2004_insol_cut`) is on a grid with 1 kyr timestep between each insolation value.

In [6]:

```
# 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 = string("Solar flux
", L"[W/m^{2}]"),
    legend = :topleft,
    grid = false,
    size = (800,200)
    )
```

```
-797.0:1.0:-13.0 - the cut version of La2004 time series
-797.0:1.0:-13.0 - is now the same as the common grid
```

Out[6]:



- 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 [7]:

```
# 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)
);
```

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

- 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 [8]:

```
# 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)
    );
```

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

---

uivD cut troubleshooting

In [7]:

```
#= 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 [10]:

```
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[10]:

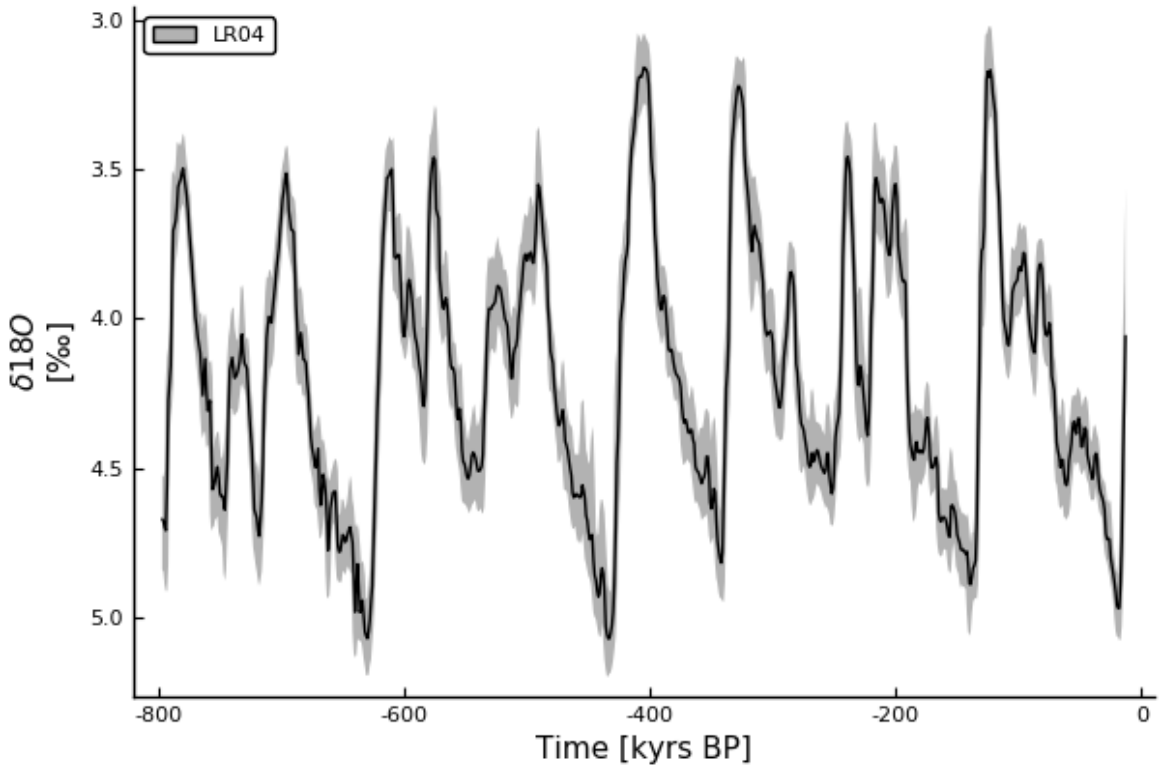
```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 785 uncertain values coupled with 785 uncertain indic
es
```

In [11]:

```
# 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
    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
    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 [kyrs BP]",
    ylabel =  string(L"\delta{18}O","
",L"[\perthousand]"),
    grid = false,
    legend = :topleft,
    yflip = true
)
```

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

Out[11]:



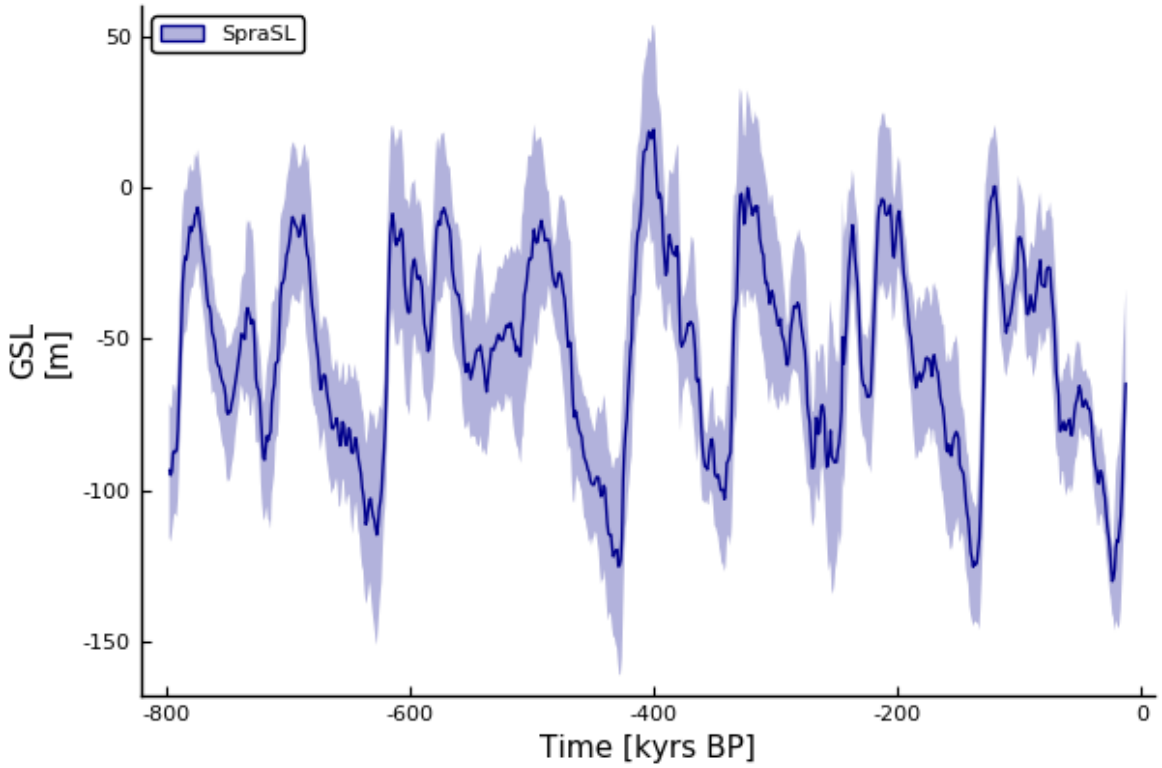
Sea level - SprattLisiecki GSL stack

In [12]:

```
ts = SL_binned_fullength_noageunc # age uncertainty already included

# 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
    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
SL_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_SL =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft,
    color = :darkblue,
    label =  "SpraSL",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "GSL
[m]",
    grid = false
    )
```

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

Out[12]:



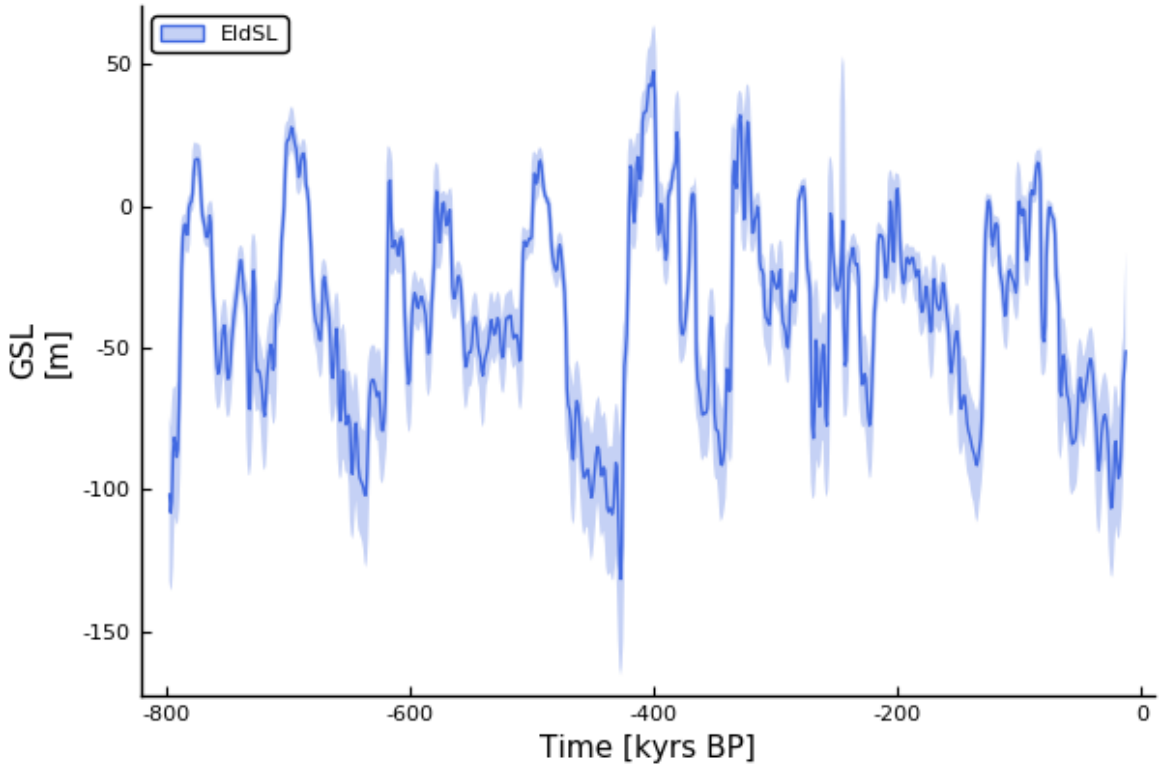
Sea level - Elderfield record

In [13]:

```
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
    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)
;

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

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

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

Out[13]:



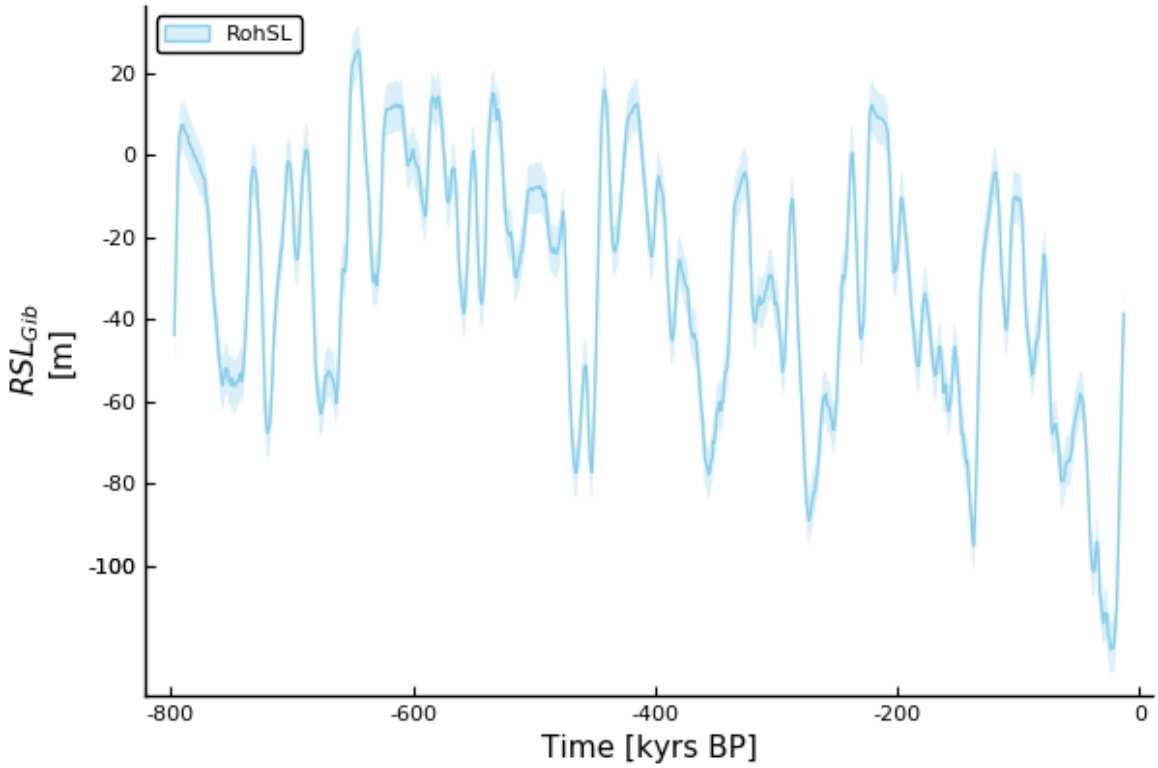
Sea level - Rohling record

In [14]:

```
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
    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 [kyrs BP]",
    ylabel =  string(L"RSL_{Gib}", "
[m]"),
    grid = false
    )
```

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

Out[14]:



---

pCO2 - Bereiter record

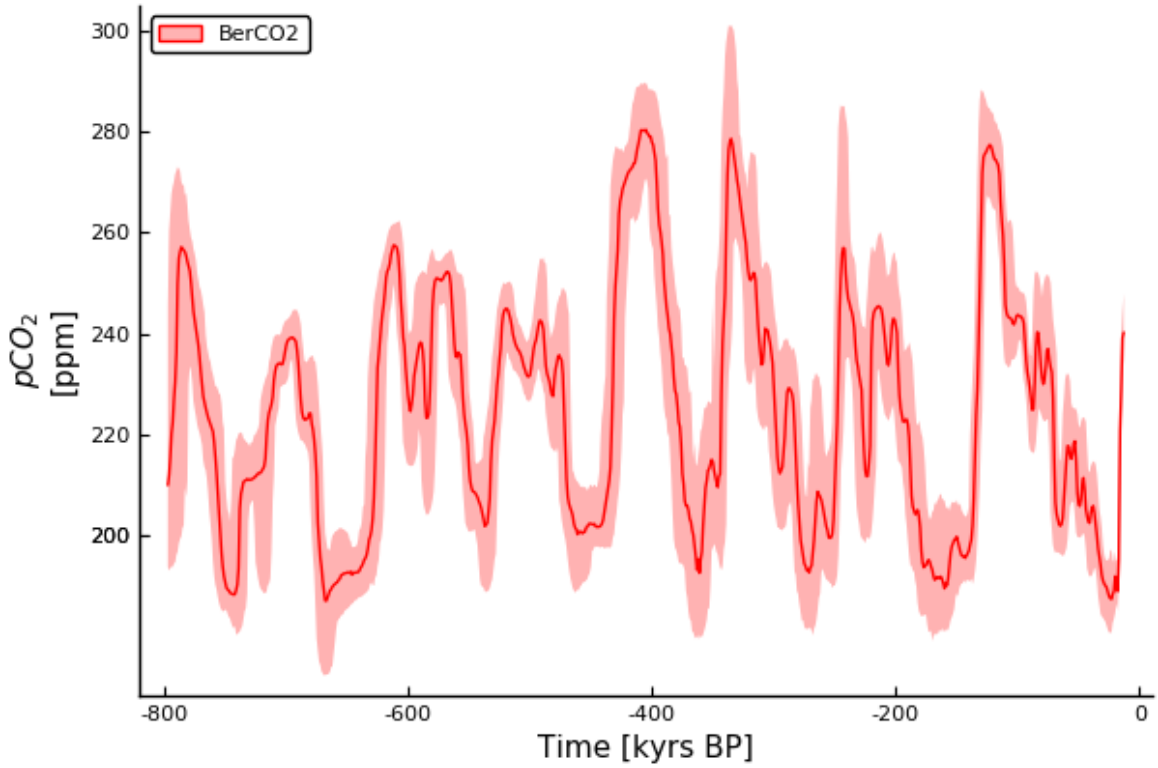
- 1 kyr grid (`B_cut`)

In [8]:

```
ts = B_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
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# save in an unambiguous name
B_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_B =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :red,
    label =  "BerCO2",
    xlabel =  "Time [kyrs BP]",
    ylabel =  string(L"pCO_{2}", "
[ppm]"),
    grid = false
    )
```

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

Out[8]:



- B_cut_hr500

In [9]:

```
# version without age uncertainty

ts = B_binned_fullength_ageunc_hr500

# 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_hr500
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# Save with an unambiguous name
B_cut_hr500 = 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_B_hr500 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :red,
    label =  "BerCO2_hr500",
    xlabel =  "Time [kyrs BP]",
    ylabel =  string(L"pCO2", " [ppm]"),
    grid = false
    )
;
```

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

We also make a version without age uncertainty, to use for analysis with the Lambert dust record (both are from Epica Dome C ice core, and can thus be analysed by depth instead of age model)

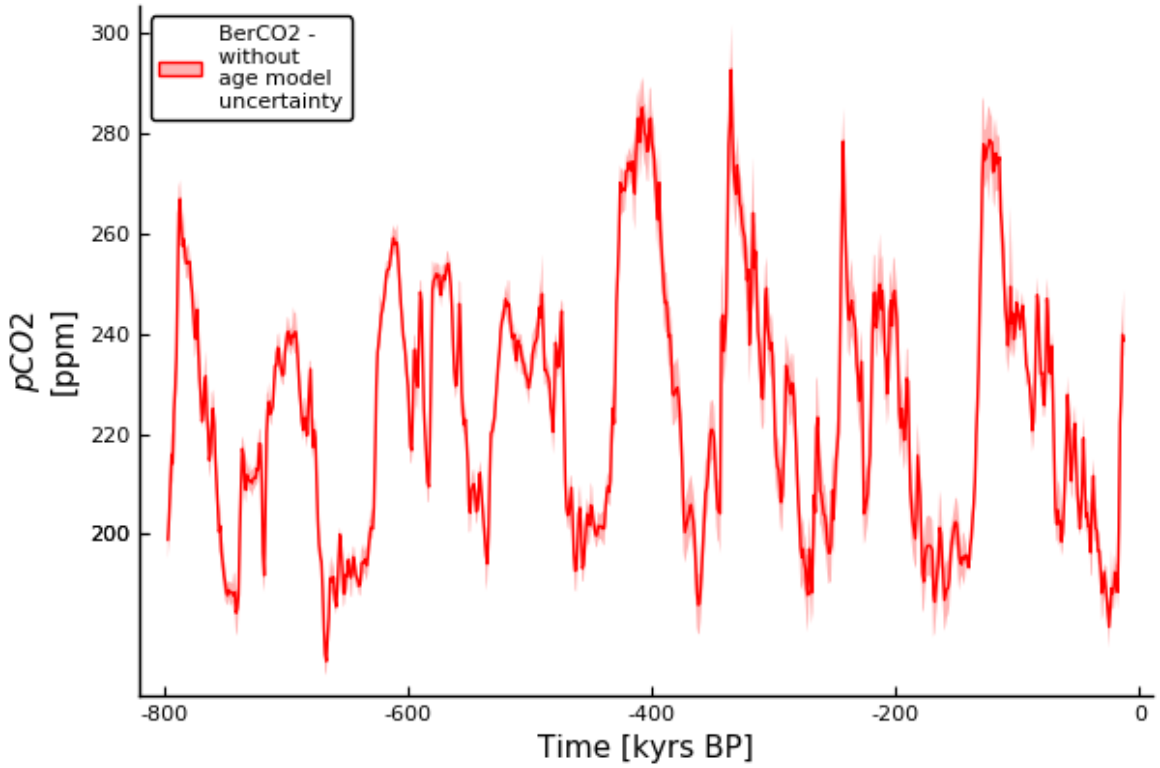
- `B_cut_edc`

In [16]:

```
# version without age uncertainty

ts = B_binned_fullength_noageuncEDC

# 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
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# Save with an unambiguous name
B_cut_edc = 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_B_edc =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :red,
    label =  "BerCO2 -
without
age model
uncertainty",
    xlabel =  "Time [kyrs BP]",
    ylabel =  string(L"pCO2","
[ppm]"),
    grid = false
    )
```

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

Out[16]:



- B_cut_edc_hr500

In [56]:

```
# version without age uncertainty

ts = B_binned_fullength_noageuncEDC_hr500

# 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_hr500
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# Save with an unambiguous name
B_cut_edc_hr500 = 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_B_edc_hr500 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :red,
    label =  "BerCO2_hr500 - without age model uncertainty",
    xlabel =  "Time [kyrs BP]",
    ylabel =  string(L"pCO2", " [ppm]"),
    grid = false
    )
;
```

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

Dust - Lambert record

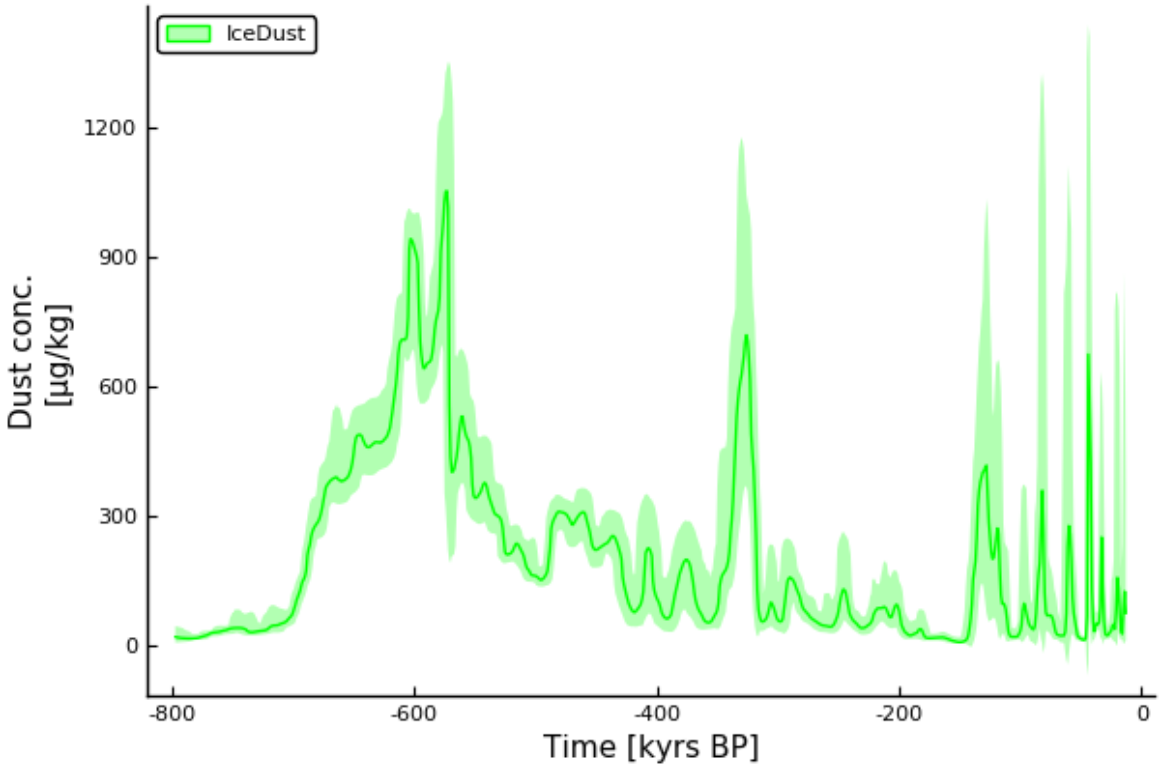
- 1 kyr grid

In [10]:

```
ts = L_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
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# Save the cut time series in an unambiguous name
L_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_L =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :lime,
    label =  "IceDust",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc.
[μg/kg]",
    grid = false
    )
```

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

Out[10]:



- L_cut_hr500

In [11]:

```
ts = L_binned_full_hr500

# 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_hr500
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# Save the cut time series in an unambiguous name
L_cut_hr500 = 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_upperq = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lowerq = 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_L_hr500 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3, legend = :topleft, color = :lime,
    label =  "IceDust_hr500",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc. [μg/kg]",
    grid = false
    );
```

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

We also prepare a version without age uncertainty, for analysis with the Bereiter pCO2 record (Both records are ice cores from the same place, Epica Dome C. This allows us to analyse by depth instead of the constructed age model, and thus avoiding the age uncertainty.

> Help: some age uncertainty in the determination of lock-in time of gas in ice SHOULD be taken into account. However, I have not found how to do this (don't understand explanation in dataset (lidie?)

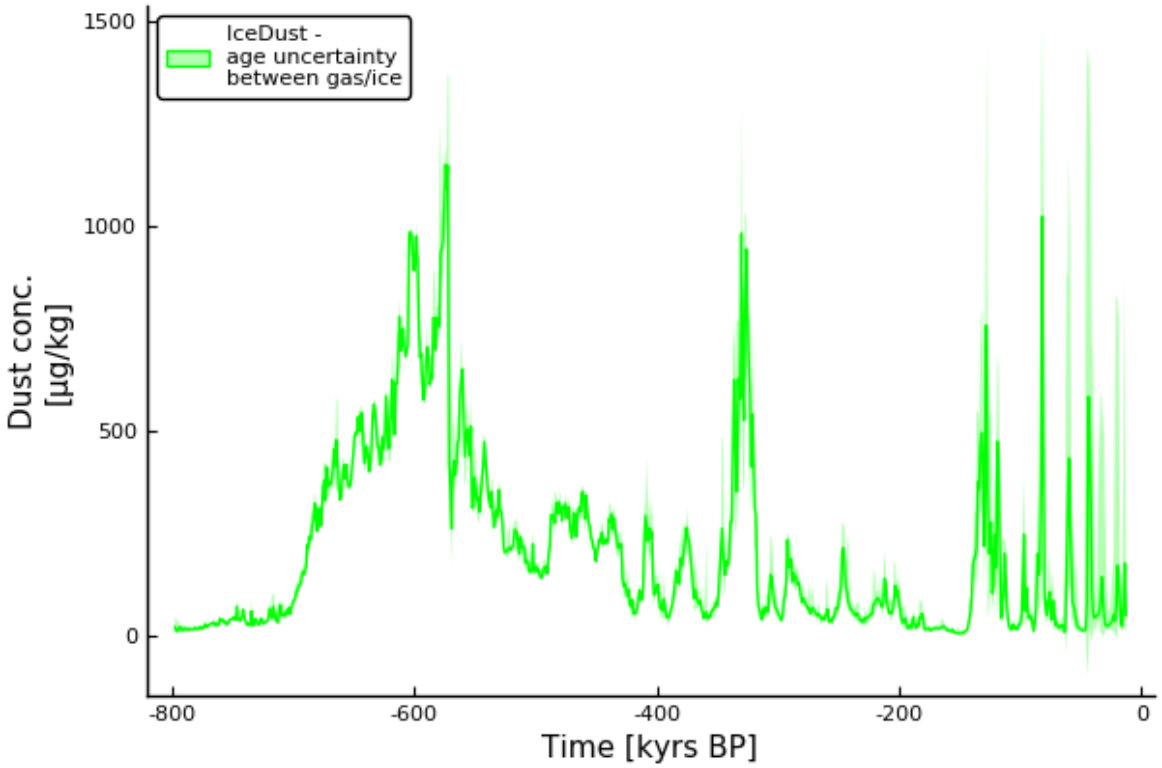
- `L_cut_edc`

In [89]:

```
ts = L_binned_full_EDC

# 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
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# Save the cut time series in an unambiguous name
L_cut_edc = 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_upperq = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lowerq = 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_L_edc =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3, legend = :topleft, color = :lime,
    label =  "IceDust -
age uncertainty
between gas/ice",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc.
[μg/kg]",
    grid = false
    )
```

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

Out[89]:



- L_cut_edc_hr500

In [90]:

```
ts = L_binned_full_hr500_edc

# 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_hr500
    print("The record is now on the common time grid")
    else
    print("Something's wrong")
    end
end

# Save the cut time series in an unambiguous name
L_cut_edc_hr500 = 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_upperq = quantile.(ts_cut.values, 0.975) .- bin_median
bin_lowerq = 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_L_edc_hr500 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3, legend = :topleft, color = :lime,
    label =  "IceDust_hr500 - age uncertainty between gas and ice",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc. [μg/kg]",
    grid = false
    );
```

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

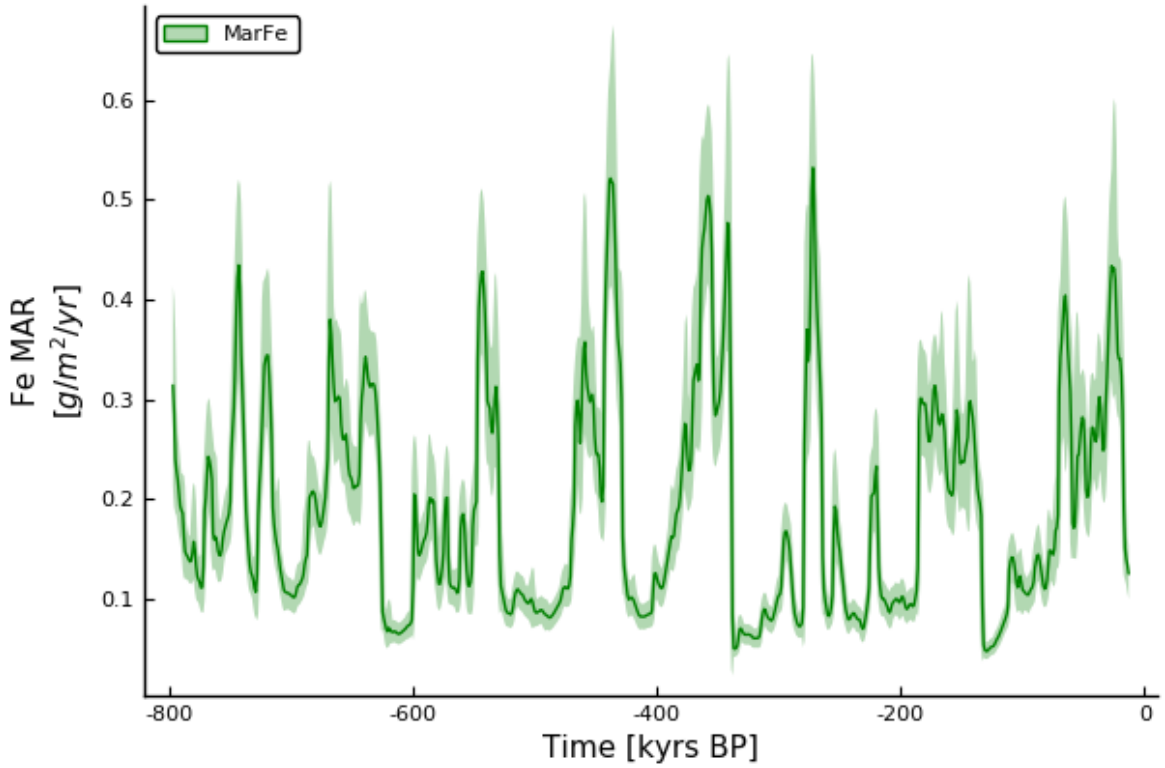
dust - Martinez-García Fe MAR record

In [91]:

```
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
    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 [kyrs BP]",
    ylabel =  string("Fe MAR","
",L"[g/m^{2}/yr]"),
    grid = false
    )
```

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

Out[91]:



We also prepare a high resolution version with timestep of 500 years, for analysis with La2004, Chalk and Grant hr records:

- `MG_cut_hr500`

In [92]:

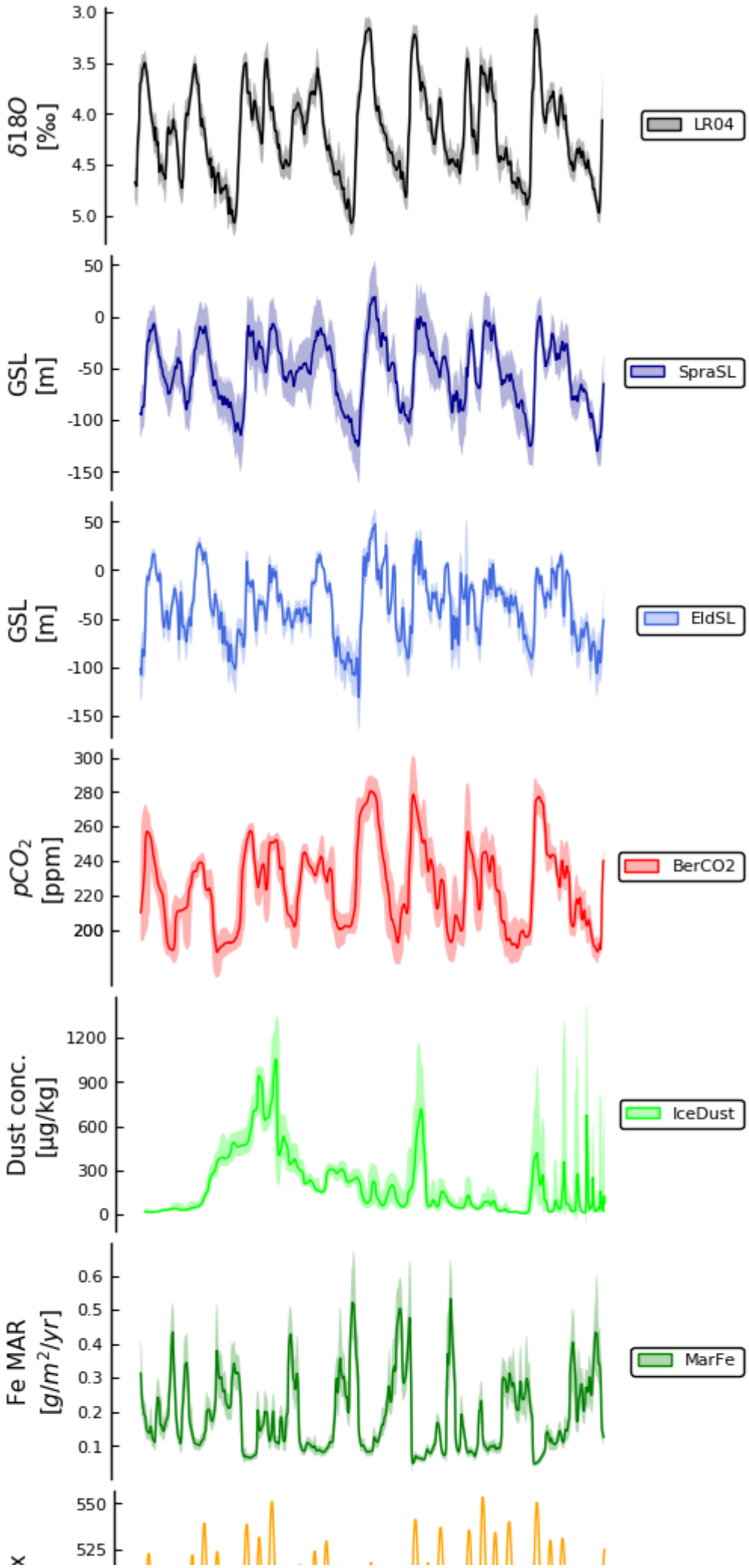
```
ts = MG_binned_postmpt_hr500

# 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_hr500
    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_hr500 = 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_MG_hr500 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :green,
    label =  "MarFe_hr500",
    xlabel =  "Time [kyrs BP]",
    ylabel =  L"Fe \ MAR \ [g/m^{2}/year]",
    grid = false
    )
;
```

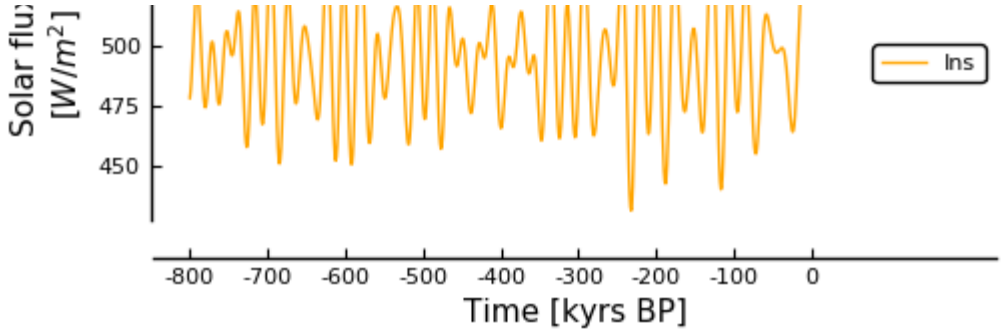
```
The record is now on the common time grid
```

Overview of time series

In [23]:

```
p1 = plot(plot_LR04, xlabel = "", xaxis = :off)
p2 = plot(plot_SL, xlabel = "", xaxis = :off)
p3 = plot(plot_E, xlabel = "", xaxis = :off)
#p4 = plot(plot_R, xlabel = "", xaxis = :off)
#p5 = plot(plot_G, xlabel = "", xaxis = :off)
p6 = plot(plot_B, xlabel = "", xaxis = :off)
p6_edc = plot(plot_B_edc, xlabel = "", xaxis = :off, color = :pink)
p7 = plot(plot_L, xlabel = "", xaxis = :off)
p7_edc = plot(plot_L_edc, xlabel = "", xaxis = :off, color = :olive)
p8 = plot(plot_MG, xlabel = "", xaxis = :off)
p9 = plot(plot_La2004, xlabel = "", xaxis = :off)
p_xaxis = plot(xlabel = "Time [kyrs BP]", xaxis = :on, yaxis = :off, grid = :off
)

A4 = (8.27*72, 11.69*72) # Dimensions of an A4 page (72 dots per inch)
l = @layout [a;b;c;d;e;f;g;h{0.01h}]

po_alltimeseries = plot(
    p1,p2,p3,p6,p7,p8,p9,p_xaxis,
    #plot_LR04, plot_SL, plot_E, plot_G, plot_B, plot_B_edc, plot_L, plot_L_edc,
plot_MG, plot_La2004,


    layout = l, #grid(12,1),
    size = (500,1200), #size = A4,
    xlims = (gridstart-50, gridend+250), xticks =(-800:100:0),
    #ylabel = "", ytickfont = false, yaxis = :off,
    legend = :right, bg_legend = :white,
    ymirror = false
    )

savefig("../../results_ePalus_ns20/timeseries/po_all_full_800.pdf")
```


δ18O [‰]
3.0
3.5
4.0
4.5
5.0
LR04
GSL [m]
50
0
-50
-100
-150
SpraSL
GSL [m]
50
0
-50
-100
-150
EldSL
pCO2 [ppm]
300
280
260
240
220
200
BerCO2
Dust conc. [µg/kg]
1200
900
600
300
0
IceDust
Fe MAR [g/m2/yr]
0.6
0.5
0.4
0.3
0.2
0.1
MarFe
550
525



Compute the predictive asymmetry between the time series

See method explained in NB2 and function defined in NB3

- Compute the predictive asymmetry between the two time series

- Normalize the results, and make a plot showing the 95% confidence interval

Outline for analyses

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

- LR04 - La2004

1. **GSL - insolation**

- Spratt & Lisiecki - La2004
- Elderfield - La2004
- Rohling - La2004
- Grant - La2004

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

- Spratt & Lisiecki - Bereiter
- Elderfield - Bereiter
- Rohling - Bereiter (..)
- Grant - Bereiter

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

- Spratt & Lisiecki - Chalk
- Elderfield - Chalk
- Rohling - Chalk
- Grant - Chalk

1. **GSL /** $\delta^{18}O$ **- dust**

- Spratt & Lisiecki - Lambert (..)
- Elderfield - Lambert (..)
- Rohling - Lambert (.....)
- Grant - Lambert (..)
- Spratt & Lisiecki - Martinez-García
- Elderfield - Martinez-García
- Rohling - Martinez-García
- Grant - Martinez-García

1. **pCO2 - dust**

- Bereiter - Lambert (!)
- Bereiter - Martinez-García (..)
- Chalk - Martinez-García

1. **pCO2 - insolation**

- Bereiter - La2004
- Chalk - La2004

1. **dust - insolation**

- Lambert - La2004
- 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.

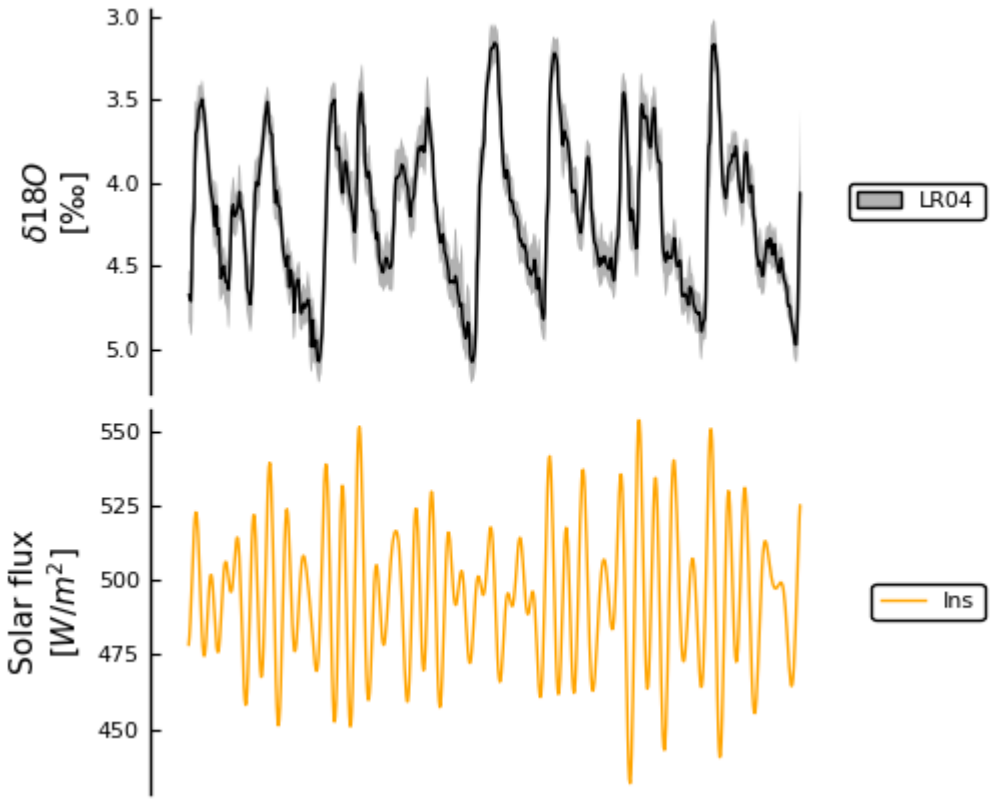
(Both the Elderfield and the Spratt-Lisiecki sea-level records are tuned to the LR04, whose age model is based on a lag of $\delta^{18}O$ behind insolation. This is based on the understanding of the the Milankovitch cycles as a key driver behind the ice age cycles, but is however, a caveat for our method, 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..)

In [24]:

```
# Plot overview of time series pair to analyse

plot_overview_LR04_La2004 =
plot(layout = grid(2,1),
    size = (500, 400),
    plot_LR04,
    plot_La2004
    )
```

Out[24]:



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

In [94]:

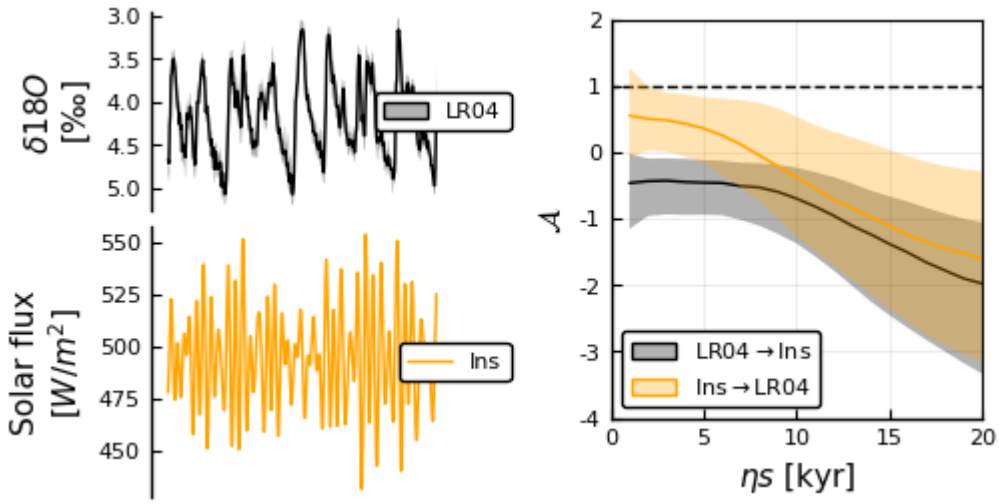
```
# Recall the time series on the common grid as X and Y
X = LR04_cut
Y = La2004_insol_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/LR04_La2004nB.j
ld2")
```

```
Results are saved in the .jld2 file.  24.095936 seconds (129.62 M al
locations: 9.466 GiB, 17.46% gc time)
```

In [25]:

```
# normalize the results for comparability

@load "../../results_ePalus_ns20/pa_jld2_files/full_800/LR04_La2004nB.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 9
5%).

### 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] .- 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
;

#Define the results plotLR04_La2004.jld2

# 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 = (-4,2),
    yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  "" # significance treshold
    )
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
```

```
    )


# Plot overview of time series
plot_overview_LR04_La2004 = plot(
    layout= grid(2,1),
    plot_LR04,
    plot_La2004)

# join plots of time series and pa results to a results subplot
plot_results_LR04_La2004 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_LR04_La2004,
    plot_normPA_LR04_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/LR04_La2004nB.pdf")
```



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.

---

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

SprattLisiecki - La2004

In [97]:

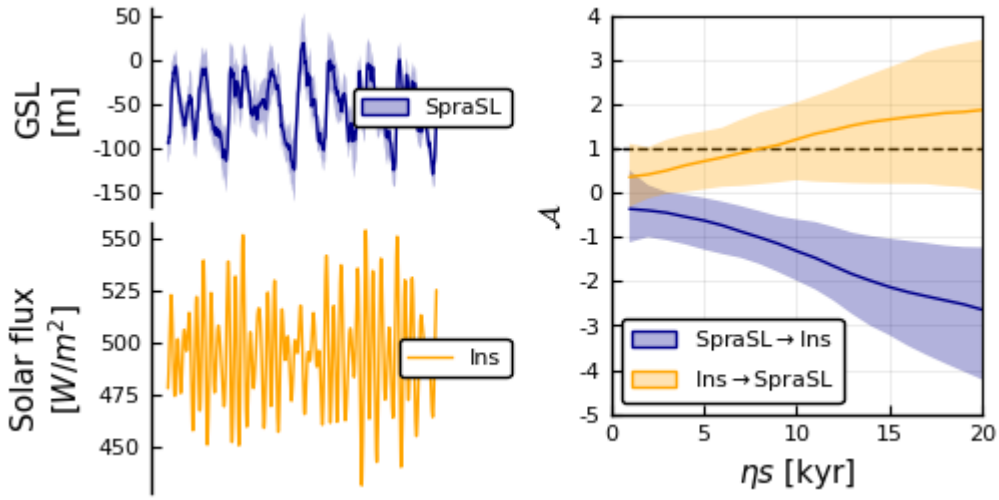
```
# First, recall the time series on the common grid as X and Y
X = SL_cut
Y = La2004_insol_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_La2004nB.jld
2")
```

```
Results are saved in the .jld2 file.  25.953708 seconds (129.61 M al
locations: 9.466 GiB, 16.70% gc time)
```

In [26]:

```
# Load and normalize the predictive asymmetry results

# Load and normalize the random sequences results of predictive asymmetry
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_La2004nB.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_SL_La2004 =
plot(#title =
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,4), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  "" # significance treshold
    )
plot!(normPA_XtoY_median, # ...from X to Y
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("SpraSL", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :darkblue
    )
plot!(normPA_YtoX_median, # ...from Y to X
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "SpraSL"),
    fillalpha = 0.3, color = :orange)
;


# join plots of time series and pa results to a results subplot
plot_overview_SL_La2004 =
plot(layout = grid(2,1),
    size = (500,400),
    plot_SL,
    plot_La2004
```

```
    )

plot_results_SL_La2004 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_SL_La2004,
    plot_normPA_SL_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/SL_La2004nB.pdf")
```



Elderfield - La2004

In [100]:

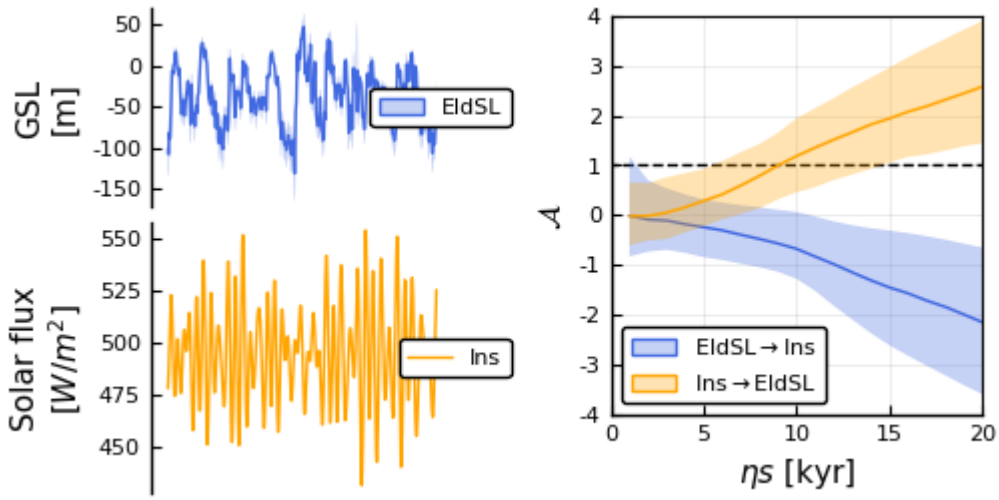
```
# Compute the predictive asymmetry between the time series

# 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, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/E_La2004.jld2"
    )
```

```
Results are saved in the .jld2 file.  24.355741 seconds (128.91 M al
locations: 9.415 GiB, 17.18% gc time)
```

In [27]:

```
# Load and normalize the predictive asymmetry results

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/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 = (-4,4),
    yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label = "" # significance treshold
    )
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 = (500,400),
    plot_E,
```

```
    plot_La2004
    )

plot_results_E_La2004 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_E_La2004,
    plot_normPA_E_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/E_La2004.pdf")
```



Rohling - La2004

In [65]:

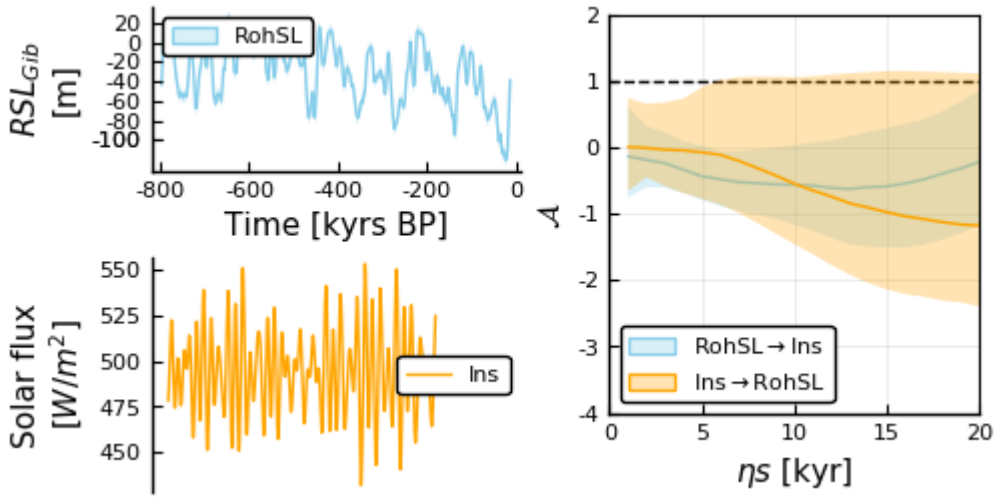
```
# Recall the time series on the common grid as X and Y
X = R_cut
Y = La2004_insol_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/R_La2004.jld2")
```

```
Results are saved in the .jld2 file.  59.311379 seconds (348.95 M al
locations: 24.906 GiB, 18.16% gc time)
```

In [28]:

```
# Load and normalize the predictive asymmetry results

# Load and normalize the results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/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 = (-4,2), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  "" # significance treshold
    )
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 = (500,250),
    layout = grid(1,2),
    plot_overview_R_La2004,
    plot_normPA_R_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/R_La2004.pdf")
```

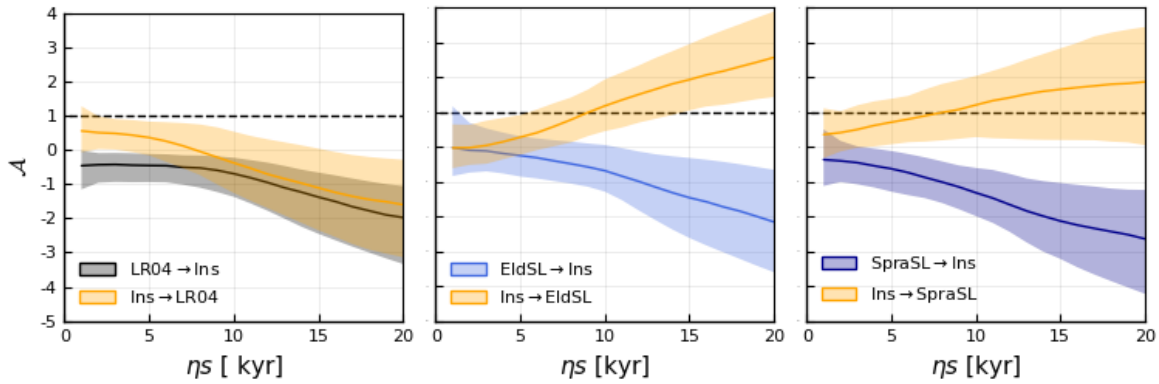


Ensemble plot of results GSL/$\delta^{18}O$ - insolation

In [29]:

```
p20 = plot(plot_normPA_LR04_La2004, ylabel = L"\mathcal{A}", ytickfont = 8, xlab
el = string(L"ηs"," [ kyr]"))
p21 = plot(plot_normPA_E_La2004, ylabel = "", ytickfont = false, xlabel = string
(L"ηs"," [kyr]"))
p22 = plot(plot_normPA_R_La2004, ylabel = "", ytickfont = false, xlabel = string
(L"ηs"," [kyr]"))
p23 = plot(plot_normPA_SL_La2004, ylabel = "", ytickfont = false, xlabel = strin
g(L"ηs"," [kyr]"))
#p24 = plot(plot_normPA_G_La2004nB, ylabel = string("GSL - insolation"), ymirror
= true, ytickfont = false, xlabel = string(L"ηs"," [ kyr]"))
#p24hr = plot(plot_normPA_G_La2004nB_hr500, ylabel = "", ytickfont = false)

pe_gsl_insol = plot(
    p20,p21,p23,#p24,
    layout = grid(1,3), size = (750,250), # without Rohling
    ylims= (-5,4), yticks = (-5:1:5),
    border = true, legend = :bottomleft, bg_legend = :transparent
    )

#savefig("../../results_ePalus_ns20/ensemble_normPA_plots/full_800/insol-gsl.pd
f")
```

Out[29]:



3. $\mathcal{A}$ between sea level / $\delta^{18}O$ and pCO2

LR04 - Bereiter

In [73]:

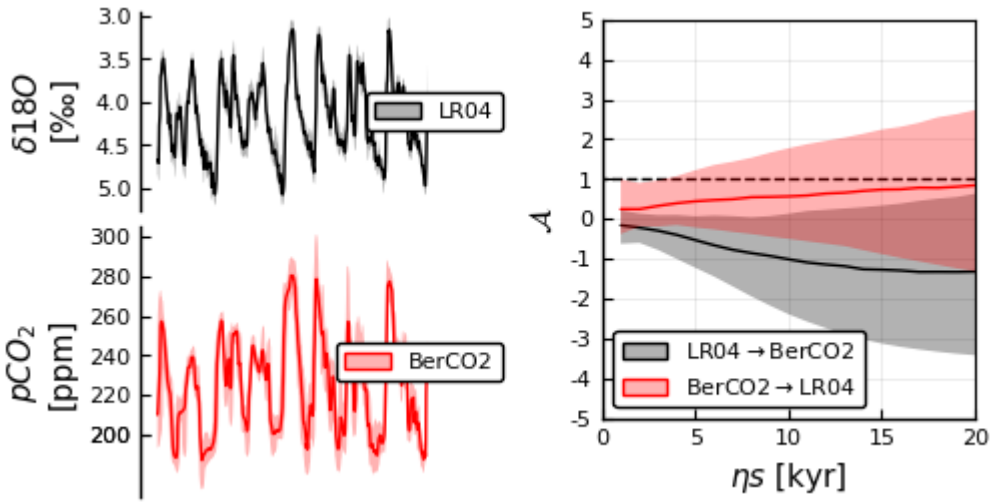
```
# recall the time series on the common grid as X and Y
X = LR04_cut
Y = B_cut

# Compute a family of transfer entropies and predictive asymmetries between the
 records (function detailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/LR04_B.jld2")
```

```
Results are saved in the .jld2 file.  63.651544 seconds (359.02 M al
locations: 25.278 GiB, 17.55% gc time)
```

In [33]:

```
# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/LR04_B.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_LR04_B =
plot(#title = string(L"$\mathcal{A}$", " between sea level and pCO2")
    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 = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("LR04", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3, color = :black
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("BerCO2", L"\rightarrow", "LR04"),
    fillalpha = 0.3, color = :red)
;


# join plots of time series and pa results to a results subplot
plot_overview_LR04_B =
plot(layout = grid(2,1),
    size = (500,400),
    plot_LR04,
    plot_B
    )
```

```
plot_results_LR04_B =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_LR04_B,
    plot_normPA_LR04_B)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/LR04_B.pdf")
```



Spratt & Lisiecki - Bereiter

In [76]:

```
# recall the time series on the common grid as X and Y
X = SL_cut
Y = B_cut

# Compute the transfer entropy and predictive asymmetry between the  records (fu
nction detailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_B.jld2")
```

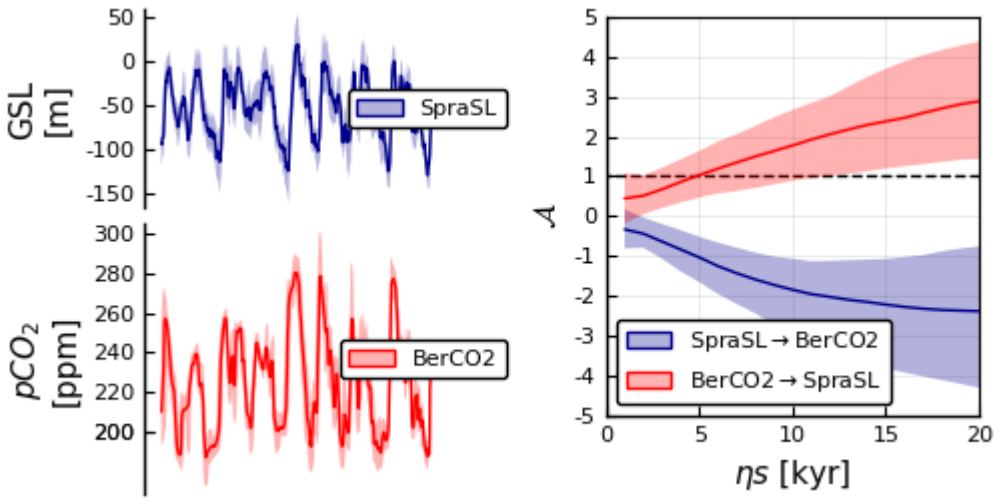
```
Results are saved in the .jld2 file.  65.556281 seconds (356.80 M al
locations: 25.123 GiB, 17.30% gc time)
```

In [34]:

```
# Load and normalize the predictive asymmetry results

@load "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_B.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_SL_B =
plot(#title = string(L"$\mathcal{A}$", " between sea level and pCO2")
    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 = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("SpraSL", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3, color = :darkblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("BerCO2", L"\rightarrow", "SpraSL"),
    fillalpha = 0.3, color = :red)
;


# join plots of time series and pa results to a results subplot
plot_overview_SL_B =
plot(layout = grid(2,1),
    size = (500,400),
    plot_SL,
    plot_B
    )
```

```
plot_results_SL_B =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_SL_B,
    plot_normPA_SL_B)

#savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/SL_B_noInterpolation.pdf")
```

Out[34]:



Elderfield - Bereiter

In [124]:

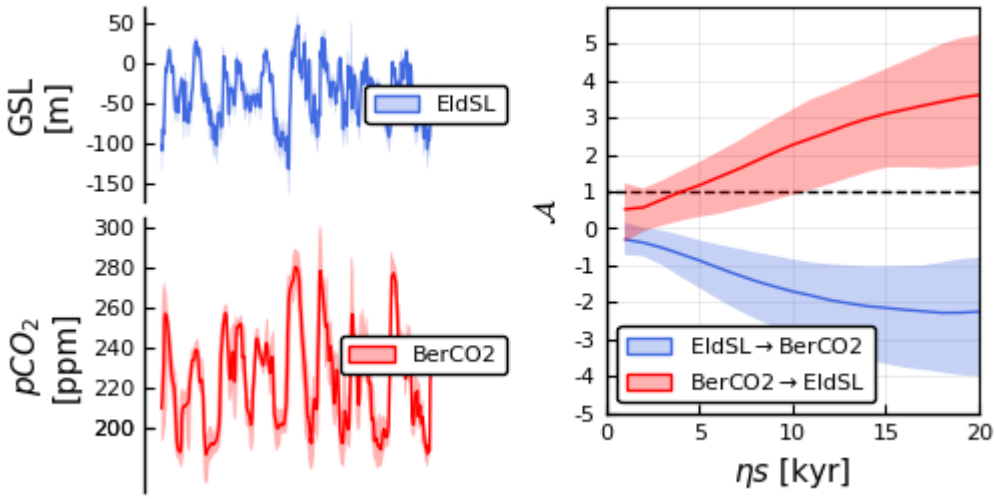
```
# recall the time series on the common grid as X and Y
X = E_cut
Y = B_cut

# Compute the transfer entropy and predictive asymmetry between the records (function detailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/E_B.jld2")
```

```
Results are saved in the .jld2 file.  61.199371 seconds (352.63 M allocations: 24.833 GiB, 17.50% gc time)
```

In [35]:

```
# Load and normalize the predictive asymmetry results

@load "../../results_ePalus_ns20/pa_jld2_files/full_800/E_B.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_B =
plot(#title = L"$\mathcal{A}$ between sea level and pCO2"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,6), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("EldSL", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3,  color = :royalblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("BerCO2", L"\rightarrow", "EldSL"),
    fillalpha = 0.3,  color = :red)
;


# join plots of time series and pa results to a results subplot
plot_overview_E_B =
plot(layout = grid(2,1),
    size = (500,400),
    plot_E,
    plot_B
```

```
    )

plot_results_E_B =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_E_B,
    plot_normPA_E_B)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/E_B.pdf")
```



Rohling - Bereiter

(NB: due to sapropelic intervals (intervals of major freshwater dilution), the Rohling record is not representative of global sea level for the past 700 kyrs, and therefore not really suited for post-MPT analyses. We compute the predictive asymmetries anyways out of curiosity, but will not include them in the work, as we kow the record is unreliable.

In [127]:

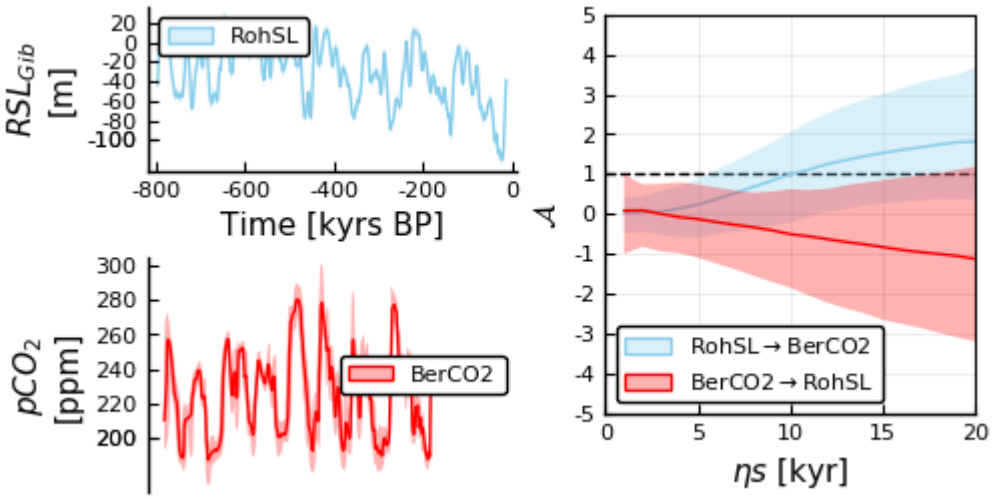
```
# recall the time series on the common grid as X and Y
X = R_cut
Y = B_cut

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/R_B.jld2")
```

```
Results are saved in the .jld2 file.  67.097908 seconds (354.99 M al
locations: 24.997 GiB, 17.83% gc time)
```

In [36]:

```
# Load and normalize the predictive asymmetry results

@load "../../results_ePalus_ns20/pa_jld2_files/full_800/R_B.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_B =
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 = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("RohSL", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3, color = :skyblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("BerCO2", L"\rightarrow", "RohSL"),
    fillalpha = 0.3, color = :red)
;


# join plots of time series and pa results to a results subplot
plot_overview_R_B =
plot(layout = grid(2,1),
    size = (500,400),
    plot_R,
    plot_B
    )
```

```
plot_results_R_B =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_R_B,
    plot_normPA_R_B)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/R_B.pdf")
```

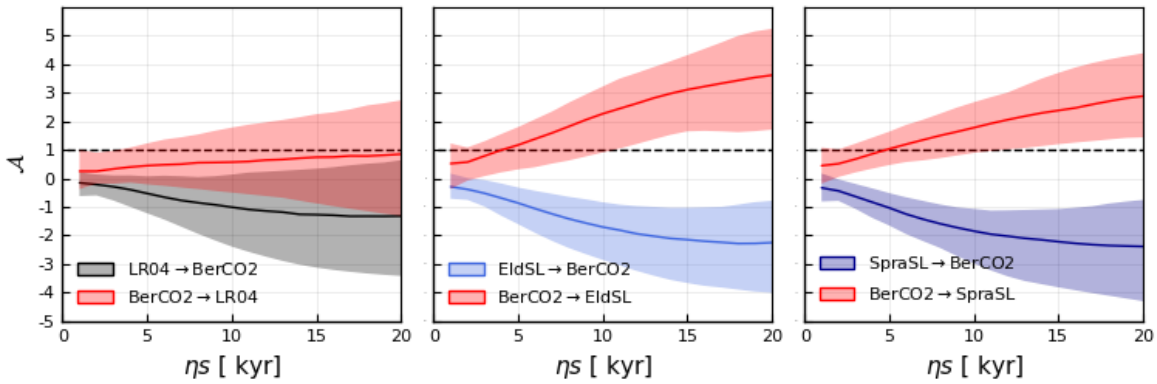


Ensemble of results GSL-pCO_2

In [37]:

```
p30 = plot( plot_normPA_LR04_B, ylabel = L"\mathcal{A}", xtickfont = 8, xlabel =
string(L"ηs"," [ kyr]"))
p31 = plot(plot_normPA_E_B, ylabel = "", ytickfont = false, xtickfont = 8, xlabe
l = string(L"ηs"," [ kyr]"))
p32 = plot(plot_normPA_SL_B, ylabel = "", ytickfont = false, xtickfont = 8, xlab
el = string(L"ηs"," [ kyr]"))
#p33 = plot(plot_normPA_R_B, ylabel = "", ytickfont = false, xtickfont = 8, xlab
el = string(L"ηs"," [ kyr]"))
#p34 = plot(plot_normPA_G_B, ylabel = string("GSL - ", L"$pCO_{2}$"), ymirror =
 true, ytickfont = false)

pe_gsl_co2 = plot(
    #p30, p31,p32,p33,p34,layout = grid(1,5),size = (1250,250),
    p30, p31, p32,#p34,
    layout = grid(1,3),size = (750,250),# without Rohling
    border = true,
    legend = :bottomleft,
    bg_legend = :transparent,
    ylims = (-5,6), yticks = (-5:1:5) )

#savefig("../../results_ePalus_ns20/ensemble_normPA_plots/full_800/gsl-co2.pdf")
```

Out[37]:



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

LR04 - Lambert

In [132]:

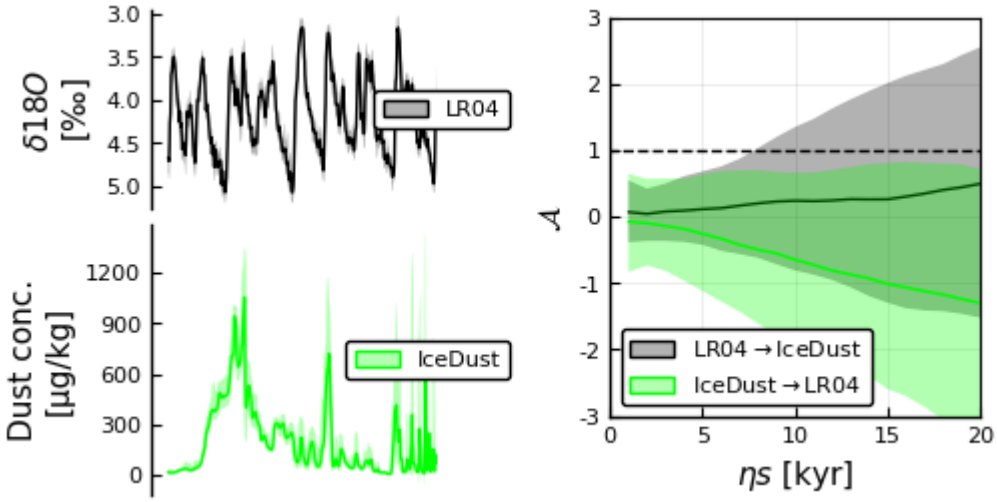
```
# recall the time series on the common grid as X and Y
X = LR04_cut
Y = L_cut

# Compute the predictive asymmetry between the two time series
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/LR04_L.jld2")
```

```
Results are saved in the .jld2 file.  64.370963 seconds (358.94 M al
locations: 25.272 GiB, 17.75% gc time)
```

In [38]:

```
# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/LR04_L.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_LR04_L =
plot(#title = L"$\mathcal{A}$ between sea level and dust over the past 800 kyr
s",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-3,3), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("LR04", L"\rightarrow", "IceDust"),
    fillalpha = 0.3,  color = :black
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "LR04"),
    fillalpha = 0.3,  color = :lime
    )


# join plots of time series and pa results to a results subplot
plot_overview_LR04_L =
plot(layout = grid(2,1),
    size = (500,400),
    plot_LR04,
    plot_L
```

```
    )

plot_results_LR04_L =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_LR04_L,
    plot_normPA_LR04_L)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/LR04_L_intp.pdf")
```



LR04 - Martinez-García

In [134]:

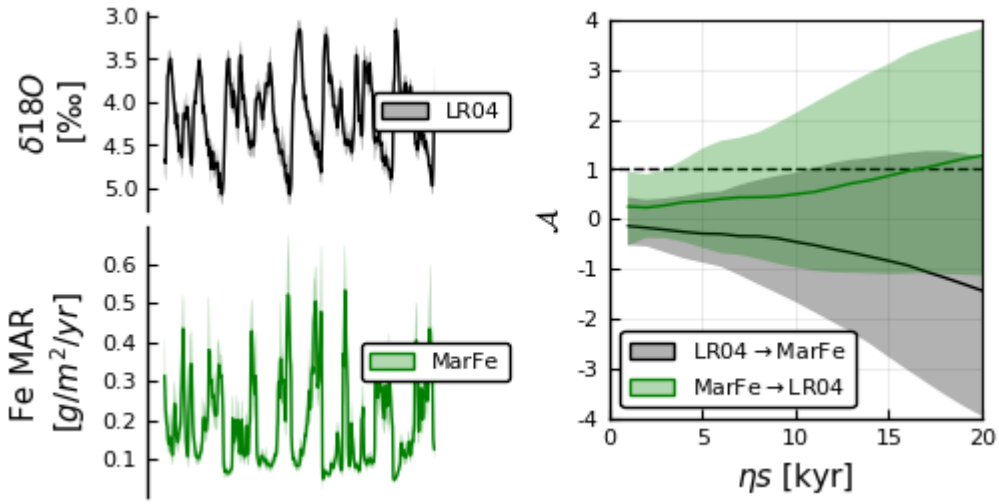
```
#Compute the predictive asymmetry between the two time series

# 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, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/LR04_MG.jld2")
```

```
Results are saved in the .jld2 file.  61.035395 seconds (356.23 M al
locations: 25.083 GiB, 19.42% gc time)
```

In [39]:

```
# Load and normalize the predictive asymmetry results

@load "../../results_ePalus_ns20/pa_jld2_files/full_800/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 = (-4,4), yticks = (-10:1:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = 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 = (500,400),
    plot_LR04,
    plot_MG
```

```
    )

plot_results_LR04_MG =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_LR04_MG,
    plot_normPA_LR04_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/LR04_MG.pdf")
```



Spratt & Lisiecki - Lambert

In [136]:

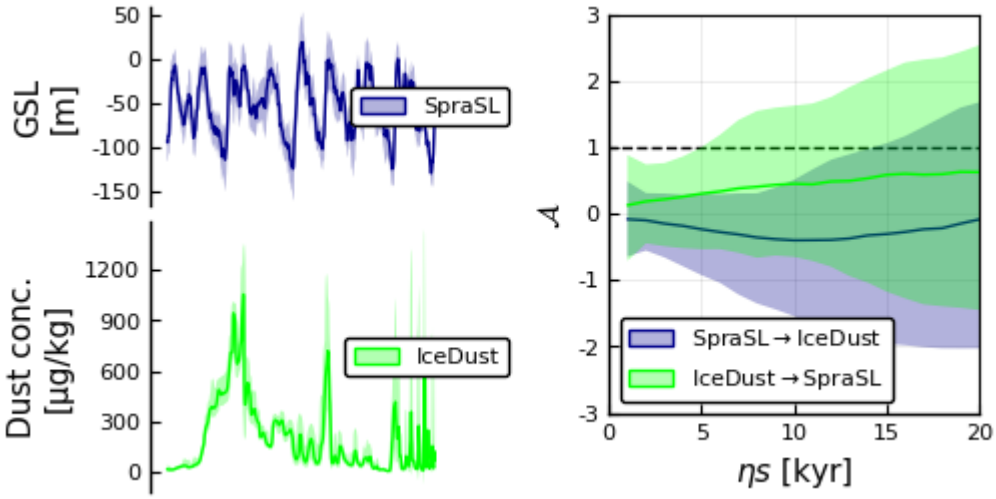
```
# recall the time series on the common grid as X and Y
X = SL_cut
Y = L_cut

# Compute the predictive asymmetry between the two time series
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_L.jld2")
```

```
Results are saved in the .jld2 file.  60.681401 seconds (352.92 M al
locations: 24.853 GiB, 18.61% gc time)
```

In [40]:

```
# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_L.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_SL_L =
plot(#title = L"$\mathcal{A}$ between sea level and dust over the past 800 kyr
s",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-3,3), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("SpraSL", L"\rightarrow", "IceDust"),
    fillalpha = 0.3,  color = :darkblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "SpraSL"),
    fillalpha = 0.3,  color = :lime
    )


# join plots of time series and pa results to a results subplot
plot_overview_SL_L =
plot(layout = grid(2,1),
    size = (500,400),
    plot_SL,
    plot_L
```

```
    )

plot_results_SL_L =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_SL_L,
    plot_normPA_SL_L)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/SL_L_intp.pdf")
```



Spratt & Lisiecki - Martinez-García

In [138]:

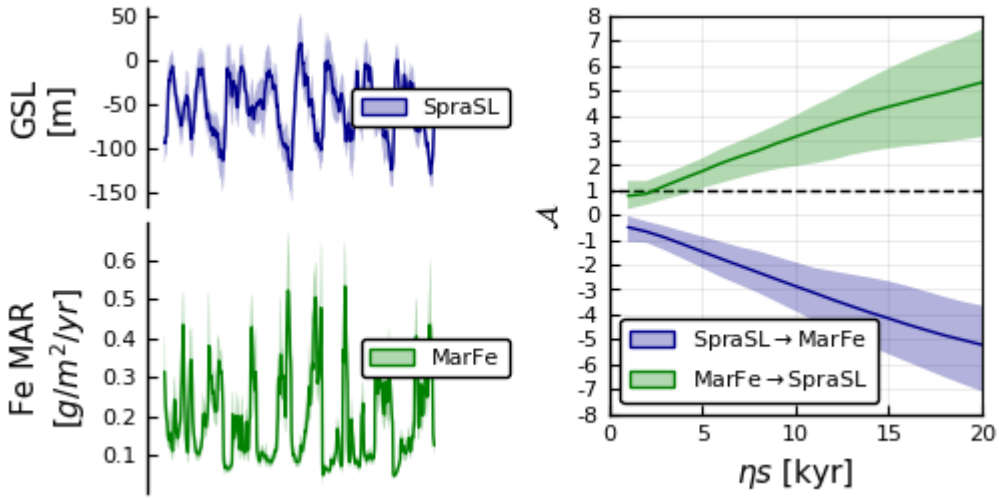
```
#Compute the predictive asymmetry between the two time series

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


# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath =  "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_MG.jld2")
```

```
Results are saved in the .jld2 file.  67.830803 seconds (356.03 M al
locations: 25.070 GiB, 17.97% gc time)
```

In [41]:

```
# Load and normalize the predictive asymmetry results


# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/SL_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_SL_MG =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-8,8), yticks = (-20:1:20),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("SpraSL", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :darkblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "SpraSL"),
    fillalpha = 0.3,  color = :green)
;


# join plots of time series and pa results to a results subplot
plot_overview_SL_MG =
plot(layout = grid(2,1),
    size = (500,400),
    plot_SL,
```

```
    plot_MG
    )

plot_results_SL_MG =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_SL_MG,
    plot_normPA_SL_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/SL_MG.pdf")
```



Elderfield - Lambert

In [140]:

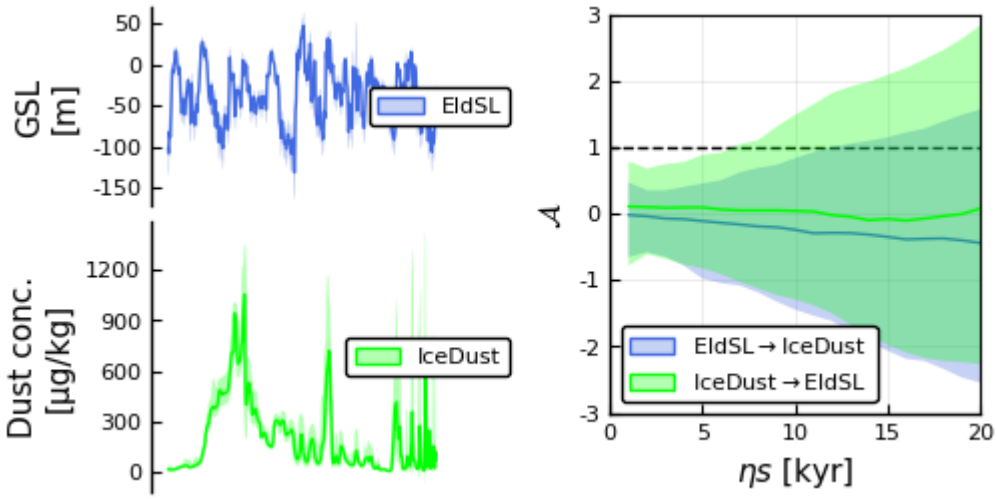
```
# recall the time series on the common grid as X and Y
X = E_cut
Y = L_cut

# Compute the predictive asymmetry between the two time series (function detaile
d in NB3)
computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/E_L.jld2")
```

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

In [42]:

```
# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/E_L.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_L =
plot(#title = L"$\mathcal{A}$ between #sea level and #dust on Antarctica"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-3,3), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("EldSL", L"\rightarrow", "IceDust"),
    fillalpha = 0.3, color = :royalblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, color = :lime)
;


# join plots of time series and pa results to a results subplot
plot_overview_E_L =
plot(layout = grid(2,1),
    size = (500,400),
    plot_E,
    plot_L
    )
```

```
plot_results_E_L =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_E_L,
    plot_normPA_E_L)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/E_L_intp.pdf")
```



Elderfield - Martinez-García

In [142]:

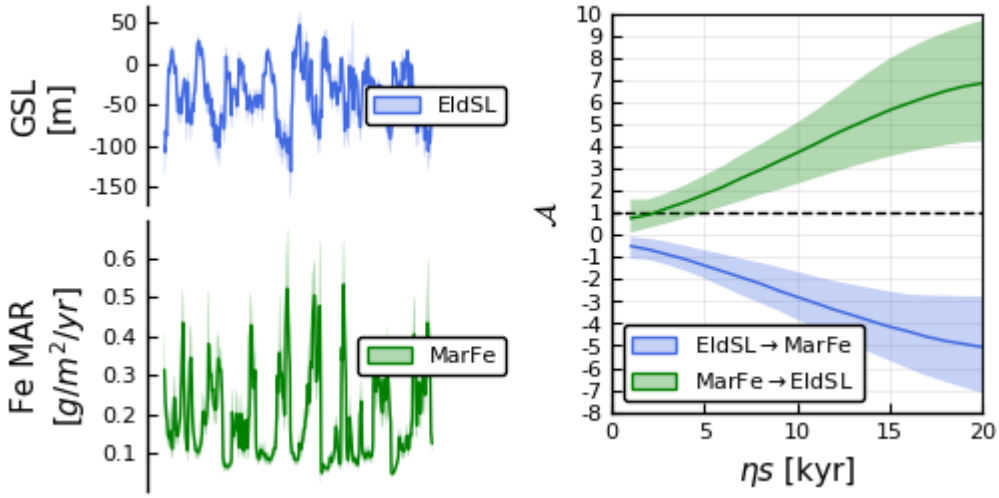
```
# recall the time series on the common grid as X and Y
X = E_cut
Y = MG_cut

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/E_MG.jld2")
```

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

In [43]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/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 = (-8,10), yticks = (-20:1:20),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = 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 = (500,400),
    plot_E,
```

```
    plot_MG
    )

plot_results_E_MG =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_E_MG,
    plot_normPA_E_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/E_MG.pdf")
```



Rohling - Lambert

Caveat, Rohling record unreliable for the post-MPT (3 sapropelic intervals identified but not removed)

In [145]:

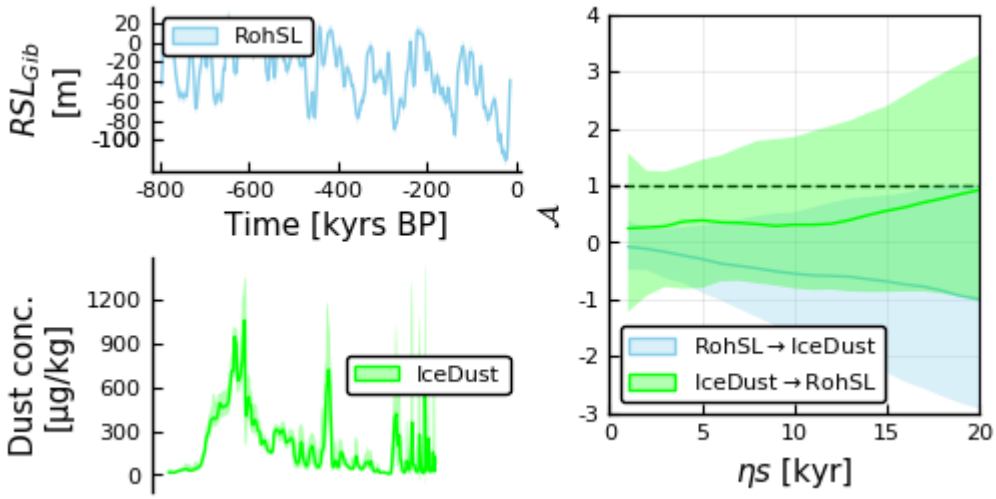
```
#Compute the predictive asymmetry between the two time series


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

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/R_L.jld2")
```

```
Results are saved in the .jld2 file.  68.915539 seconds (355.28 M al
locations: 25.017 GiB, 17.89% gc time)
```

In [44]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/R_L.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_L =
plot(xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-3,4), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("RohSL", L"\rightarrow", "IceDust"),
    fillalpha = 0.3, color = :skyblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "RohSL"),
    fillalpha = 0.3, color = :lime,
    )
;


# join plots of time series and pa results to a results subplot
plot_overview_R_L =
plot(layout = grid(2,1),
    size = (500,400),
    plot_R,
    plot_L
```

```
    )

plot_results_R_L =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_R_L,
    plot_normPA_R_L)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/R_L_intp.pdf")
```



Rohling - Martinez-García

In [154]:

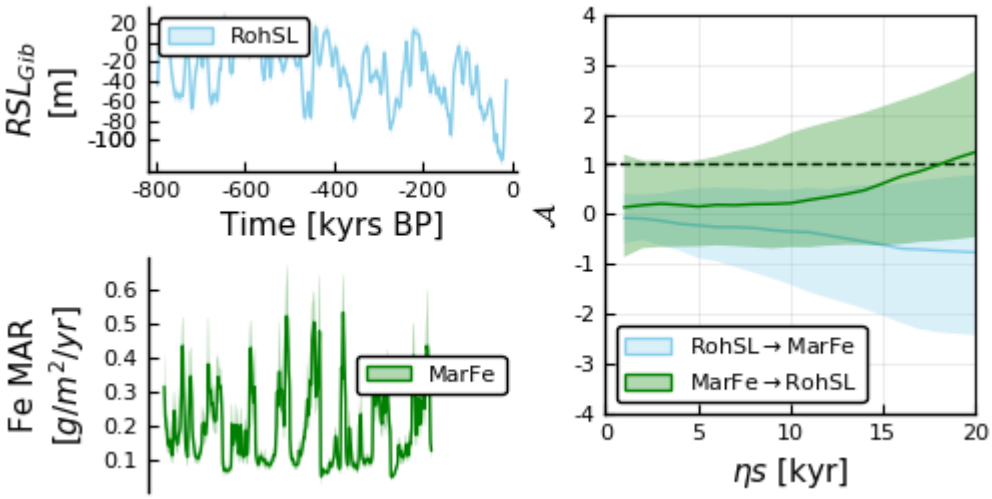
```
#Compute the predictive asymmetry between the two time series

# 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, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/R_MG.jld2")
```

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

In [45]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/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 = (-4,4), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = 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 = (500,400),
    plot_R,
    plot_MG
```

```
    )

plot_results_R_MG =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_R_MG,
    plot_normPA_R_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/R_MG.pdf")
```

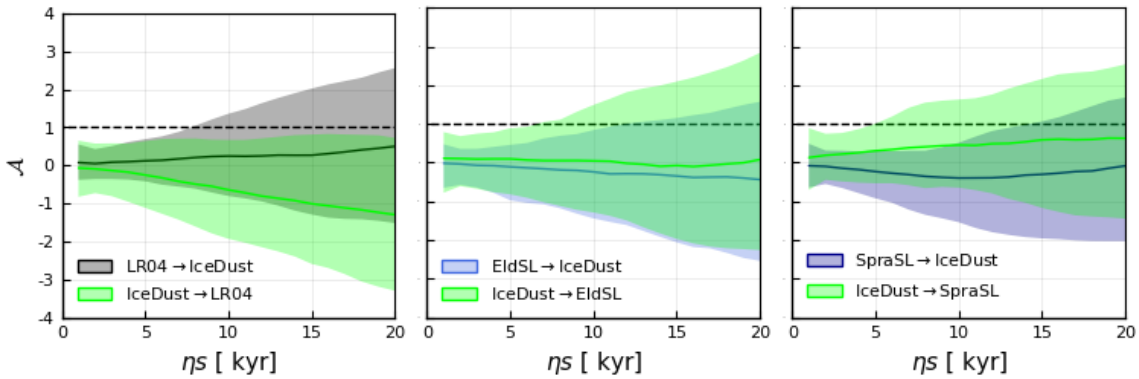


Ensemble plot of GSL-dust results

GSL - EDC dust concentration (IceDust)

In [46]:

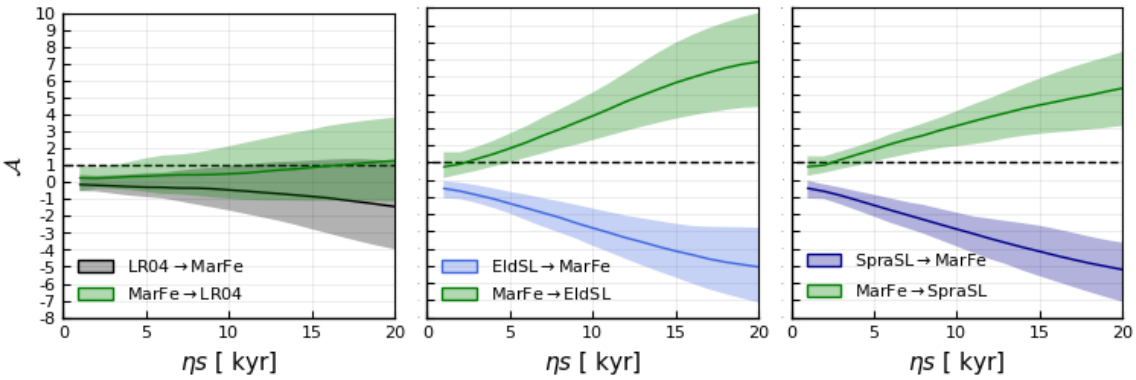
```
# GSL - IceDust (L)
p0 = plot(plot_normPA_LR04_L, ylabel = L"\mathcal{A}", xtickfont = 8, xlabel = s
tring(L"ηs"," [ kyr]"))
p1 = plot(plot_normPA_E_L, ylabel = "", ytickfont = false, xtickfont = 8, xlabel
= string(L"ηs"," [ kyr]"))
p2 = plot(plot_normPA_SL_L, ylabel = "", ytickfont = false, xtickfont = 8, xlabe
l = string(L"ηs"," [ kyr]"))
#p3 = plot(plot_normPA_R_L, ylabel = "", ytickfont = false, xtickfont = 8, xlabe
l = string(L"ηs"," [ kyr]"))
#p4 = plot(plot_normPA_G_L, xlabel = string(L"ηs", " [kyr]"),xtickfont = 8, ylab
el = string("GSL - EDC dust conc."), ymirror = true, ytickfont = false)

##### only 1 kyr grid

pe_gsl_dustL = plot(
    #p0, p1, p2, p3, p4, layout = grid(1,5), size = (1250,250),
    p0, p1, p2, layout = grid(1,3), size = (750,250), #without Rohling
    border = true,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-4,4), yticks = (-10:1:10)
    )

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/full_800/gsl-dustL_int
p.pdf")
```



GSL - Southern Ocean Fe flux (MarFe)

In [47]:

```
##### 1 kyr grid

# GSL - MarFe (MG)
p10 = plot(plot_normPA_LR04_MG, ylabel = L"\mathcal{A}", xtickfont = 8, xlabel =
string(L"ηs"," [ kyr]"))
p11 = plot(plot_normPA_E_MG, ylabel = "", ytickfont = false , xtickfont = 8, xla
bel = string(L"ηs"," [ kyr]"))
p12 = plot(plot_normPA_SL_MG, ylabel = "", ytickfont = false,  xtickfont = 8, xl
abel = string(L"ηs"," [ kyr]"))
#p13 = plot(plot_normPA_R_MG, ylabel = "", ytickfont = false,  xtickfont = 8, xl
abel = string(L"ηs"," [ kyr]"))
#p14 = plot(plot_normPA_G_MG, ylabel = string("GSL - Fe flux"), ymirror = true,
 ytickfont = false, xtickfont = 8, xlabel = string(L"ηs"," [ kyr]"))


pe_gsl_dustMG = plot(
    #p10, p11,p12,p13,p14,layout = grid(1,5), size = (1250,250),
    p10, p11,p12,layout = grid(1,3), size = (750,250), # without Rohling
    border = true, legend = :bottomleft, bg_legend = :transparent,
    ylims = (-8,10), yticks = (-10:1:10))
savefig("../../results_ePalus_ns20/ensemble_normPA_plots/full_800/e_gsl-dustMG_.
pdf")
```



5. pCO2 - dust

Bereiter - Lambert

Recall, both these time series are ice core records from Epica Dome C on West Antarctica. We will therefore use the records without time uncertainty (base time on depth rather than the developed age model).

> Note: age uncertainty due to lock-in time, ignorable? Check age model tuning in NB1.
>
> - Lock-in time typically 50 years? our high resoultuion analysis is in 125 yr bins - lock-in uncertainty ignorable or not?
> - Tired right now and don't understand, think about how this works later

- 1 kyr time step

In [162]:

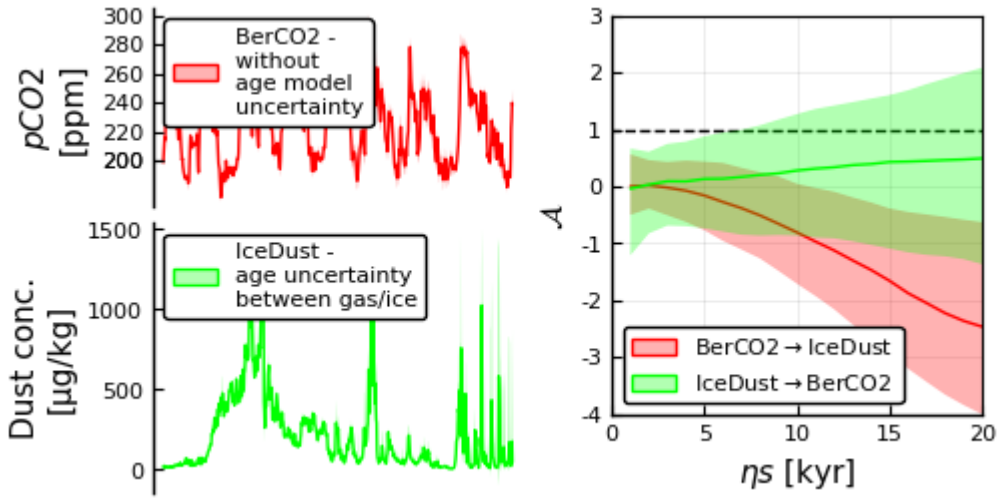
```
#Compute the predictive asymmetry between the two time series


# recall the time series on the common grid as X and Y
X = B_cut_edc
Y = L_cut_edc
binsize = 1 # recall, both time series were binned on a regular grid with timest
eps of 1 kyr (see NB1)


# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/B_L_edc.jld2")
```

```
Results are saved in the .jld2 file.  61.483565 seconds (356.57 M al
locations: 25.107 GiB, 19.84% gc time)
```

In [48]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/B_L_edc.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_B_L_edc =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-4,3), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("BerCO2", L"\rightarrow", "IceDust"),
    fillalpha = 0.3, color = :red
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3,  color = :lime)
;


# join plots of time series and pa results to a results subplot
plot_overview_B_L_edc =
plot(layout = grid(2,1),
    size = (500,400),
```

```
    plot_B_edc,
    plot_L_edc
    )

plot_results_B_L_edc =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_B_L_edc,
    plot_normPA_B_L_edc)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/B_L_edc.pdf")
```



out of curiosity, let's have a look at what predictive asymmetry we would get when using a higher age model uncertainty

In [15]:

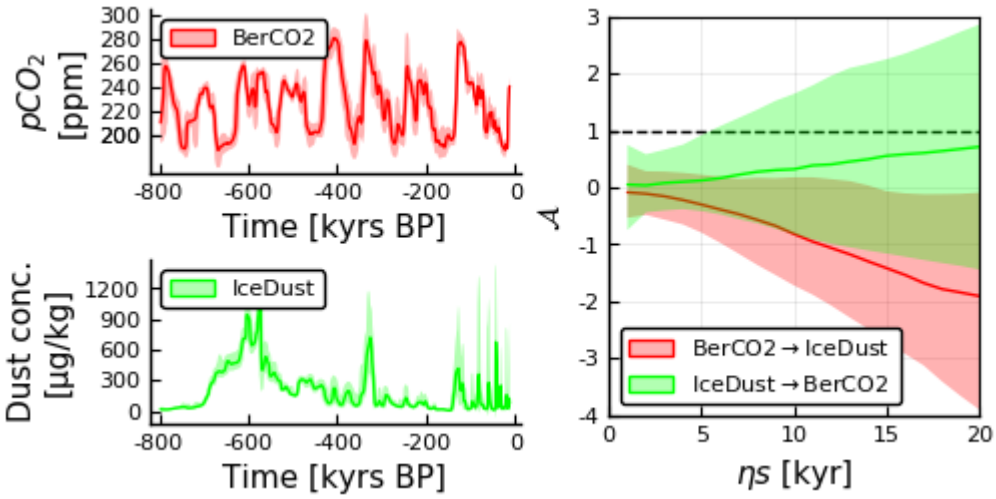
```
#Compute the predictive asymmetry between the two time series


# recall the time series on the common grid as X and Y
X = B_cut
Y = L_cut
binsize = 1 # recall, both time series were binned on a regular grid with timest
eps of 1 kyr (see NB1)


# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/B_L_HighAgeUnc.
jld2")
```

```
Results are saved in the .jld2 file.  60.641877 seconds (356.64 M al
locations: 25.112 GiB, 18.55% gc time)
```

In [16]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/B_L_HighAgeUnc.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_B_L_HighAgeUnc =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-4,3), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("BerCO2", L"\rightarrow", "IceDust"),
    fillalpha = 0.3, color = :red
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3,  color = :lime)
;


# join plots of time series and pa results to a results subplot
plot_overview_B_L_HighAgeUnc =
plot(layout = grid(2,1),
    size = (500,400),
```

```
    plot_B,
    plot_L
    )

plot_results_B_L_HighAgeUnc =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_B_L_HighAgeUnc,
    plot_normPA_B_L_HighAgeUnc)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/B_L_HighAgeUnc.pdf")
```



We see that the confidence interval widens slightly when using a higher age model uncertainty for the time series, but the qualitative results remain the same.

Sensitivity analysis using higher resolution

- 500 year timestep

In [165]:

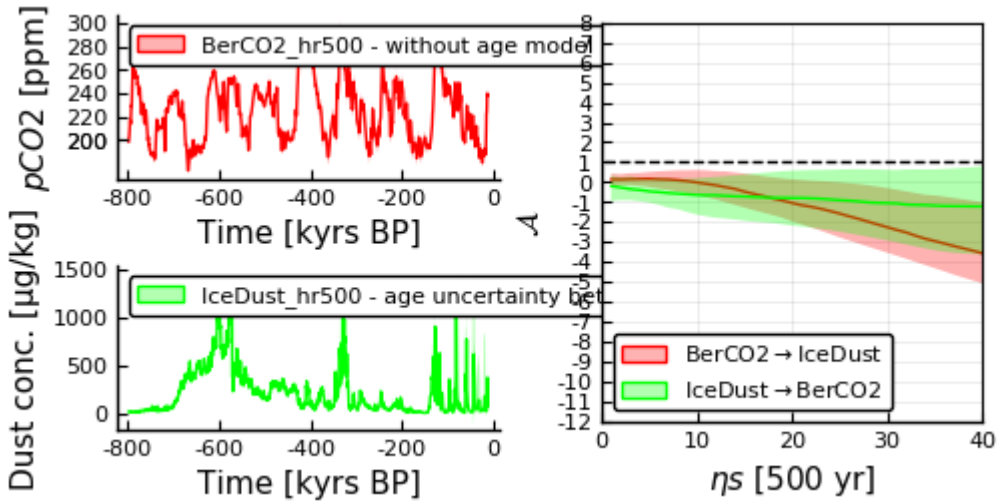
```
#Compute the predictive asymmetry between the two time series


# recall the time series on the common grid as X and Y
X = B_cut_edc_hr500
Y = L_cut_edc_hr500

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 0.500, ηmax = Int(20/0.500),
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/B_L_edc_hr500.j
ld2")
```

```
Results are saved in the .jld2 file. 247.503590 seconds (1.41 G allo
cations: 98.397 GiB, 18.69% gc time)
```

In [50]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/B_L_edc_hr500.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_B_L_edc_hr500 =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 2*5 : ηmax),
    ylims = (-12,8), yticks = (-20:1:20),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [500 yr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("BerCO2", L"\rightarrow", "IceDust"),
    fillalpha = 0.3, color = :red
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3,  color = :lime)
;


# join plots of time series and pa results to a results subplot
plot_overview_B_L_edc_hr500 =
plot(layout = grid(2,1),
    size = (500,400),
    plot_B_edc_hr500,
```

```
    plot_L_edc_hr500
    )

plot_results_B_L_edc_hr500 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_B_L_edc_hr500,
    plot_normPA_B_L_edc_hr500
)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/B_L_edc_hr500.pdf")
```



Bereiter - Martinez-García

In [52]:

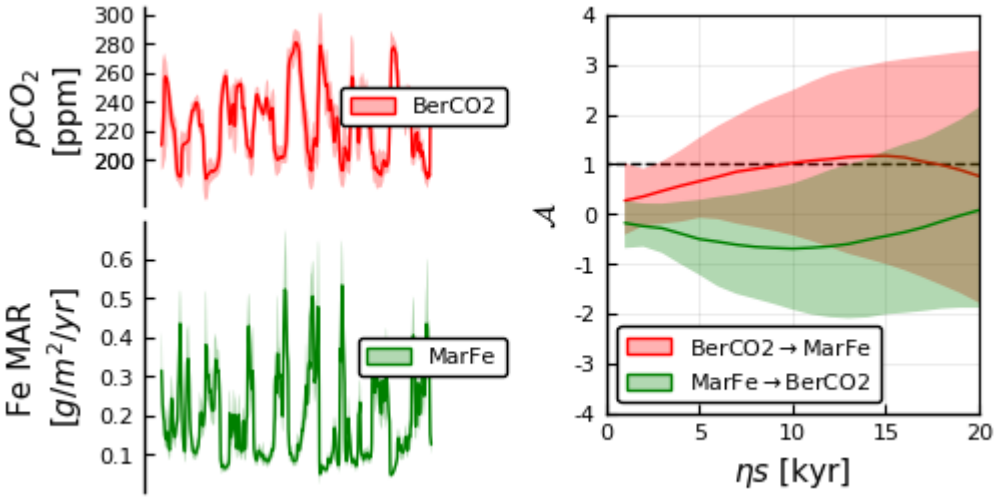
```
# recall the time series on the common grid as X and Y
X = B_cut
Y = MG_cut

# Compute the predictive asymmetry between the two time series (function detaile
d in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/B_MG.jld2")
```

```
Results are saved in the .jld2 file.  64.981387 seconds (356.81 M al
locations: 25.124 GiB, 18.24% gc time)
```

In [53]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/B_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_B_MG =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-4,4), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("BerCO2", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :red
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3, color = :green)
;


# join plots of time series and pa results to a results subplot
plot_overview_B_MG =
plot(layout = grid(2,1),
    size = (500,400),
    plot_B,
```

```
    plot_MG
    )

plot_results_B_MG =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_B_MG,
    plot_normPA_B_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/B_MG.pdf")
```



- higher resolution analyisis (500 yr timestep)

In [58]:

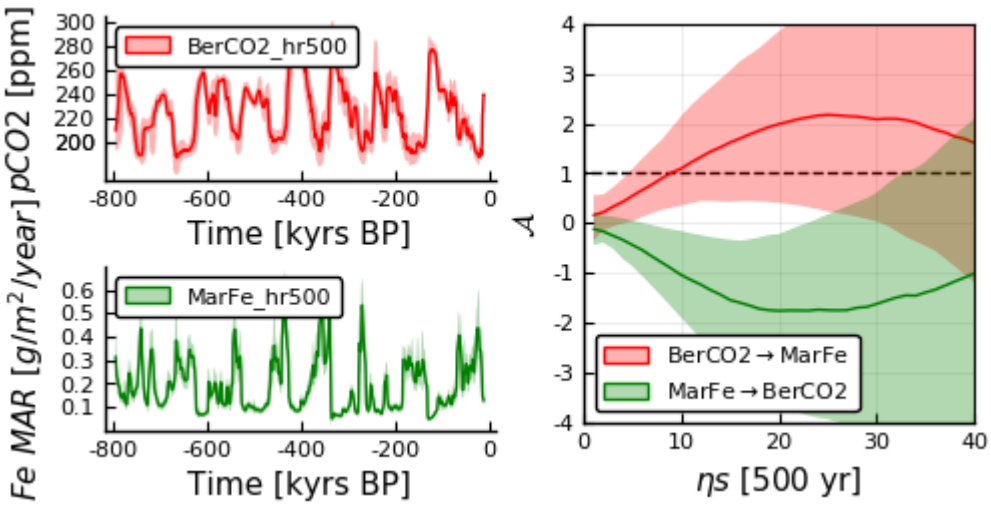
```
#Compute the predictive asymmetry between the two time series


# recall the time series on the common grid as X and Y
X = B_cut_hr500
Y = MG_cut_hr500

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 0.5, ηmax = Int(20/0.500),
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/B_MG_hr500.jld
2")
```

```
Results are saved in the .jld2 file. 245.943862 seconds (1.41 G allo
cations: 98.979 GiB, 18.27% gc time)
```

In [59]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/B_MG_hr500.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_B_MG_hr500 =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 2*5 : ηmax),
    ylims = (-4,4), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [500 yr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("BerCO2", L"\rightarrow", "MarFe"),
    fillalpha = 0.3, color = :red
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3, color = :green)
;


# join plots of time series and pa results to a results subplot
plot_overview_B_MG_hr500 =
plot(layout = grid(2,1),
    size = (500,400),
    plot_B_hr500,
```

```
    plot_MG_hr500
    )

plot_results_B_MG_hr500 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_B_MG_hr500,
    plot_normPA_B_MG_hr500)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/B_MG_hr500.pdf")
```

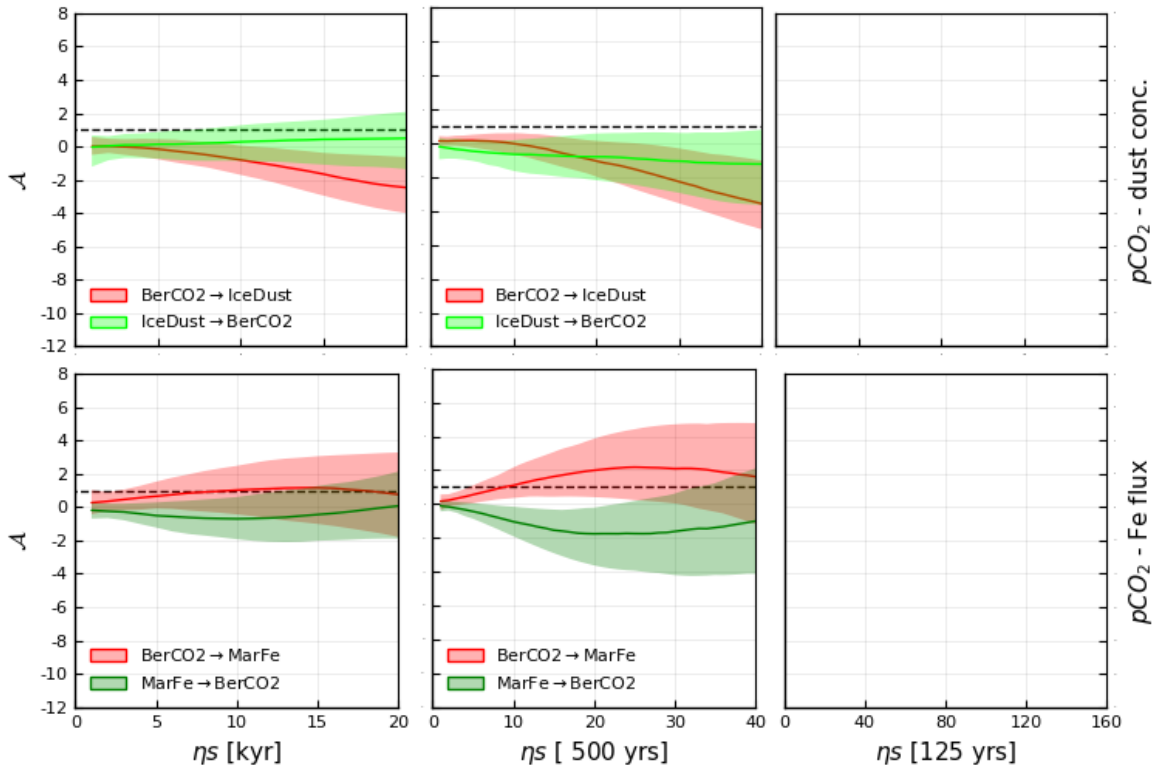


Ensemble plot of dust-pCO_2 results

In [61]:

```
# pe_co2_dust


#### BerCO2-IceDust
p51 = plot(plot_normPA_B_L_edc, ylabel = L"\mathcal{A}", ymirror = false, ytickf
ont = 8)
p52 = plot(plot_normPA_B_L_edc_hr500, ylabel = "", ytickfont = false)
p_empty1 = plot(xlims = (0,160), xticks = (0:5*8:160), ytickfont = false,
    xlabel = string(L"ηs", " [125 yrs]"),
    ymirror = true, ylabel = string(L"pCO_{2}", " - dust conc."))

pe_co2_dustL_hr = plot(p51, p52,  p_empty1, # Bereiter - Lambert
    layout = grid(1,3), xlabel = "", xtickfont = false)

#### BerCO2-MarFe

p53 = plot(plot_normPA_B_MG, ytickfont = 8, ylabel = L"\mathcal{A}")
p54 = plot(plot_normPA_B_MG_hr500, ylabel = "", ytickfont = false, xtickfont = 8
, xlabel = string(L"ηs"," [ 500 yrs]"))
p_empty2 = plot(xlims = (0,160), xticks = (0:5*8:160), ytickfont = false,
    xlabel = string(L"ηs", " [125 yrs]"),
    ymirror = true, ylabel = string(L"pCO_{2}", " - Fe flux")
    )

pe_co2_dustMG_hr = plot( p53, p54,  p_empty2, # Bereiter - Martinez-García
    layout = grid(1,3),)

#### join subplots
pe_co2_dust_hr = plot(
    pe_co2_dustL_hr,
    pe_co2_dustMG_hr,
    layout = grid(2,1), size = (750,500),
    border = true,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-12,8), yticks = (-20:2:20)
    )

#savefig("../../results_ePalus_ns20/ensemble_normPA_plots/full_800/co2-dust_hr.p
df")
#savefig("../../results_ePalus_ns20/sensitivity_analyses/hr_full_800_co2-dust_h
r.pdf")
```

Out[61]:



6. $\mathcal{A}$ between insolation and pCO2

La2004 - Bereiter

- 1 kyr timestep

In [64]:

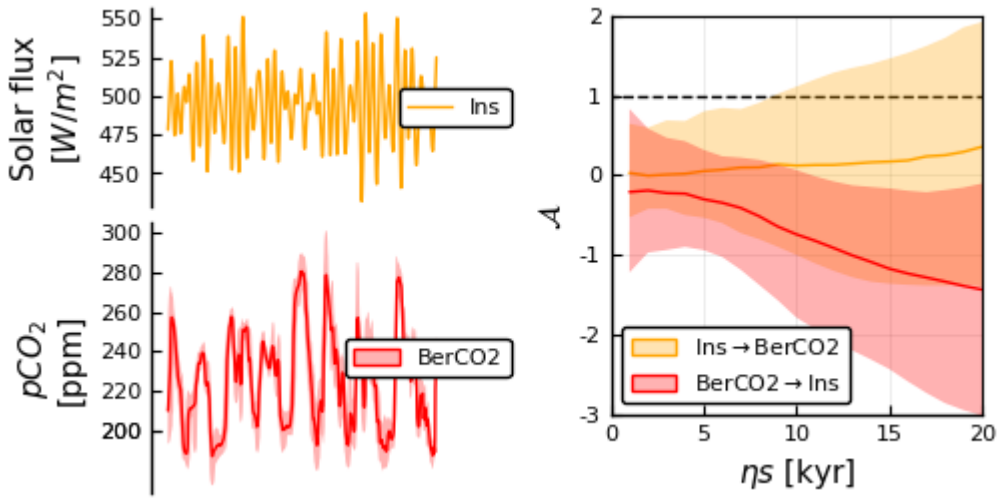
```
#Compute the predictive asymmetry between the two time series


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

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_B.jld
2")
```

```
Results are saved in the .jld2 file.  56.756016 seconds (348.87 M al
locations: 24.895 GiB, 19.48% gc time)
```

In [65]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_B.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_B =
plot(#title = L"$\mathcal{A}$ between pCO2 and insolation",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-3,2), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("Ins", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("BerCO2", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :red
    )


# join plots of time series and pa results to a results subplot
plot_overview_La2004_B =
plot(layout = grid(2,1),
    size = (500,400),
    plot_La2004,
```

```
    plot_B
    )

plot_results_La2004_B =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_La2004_B,
    plot_normPA_La2004_B)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/La2004nB_B.pdf")
```



- high resolution analysis (500 yr time step)

In [66]:

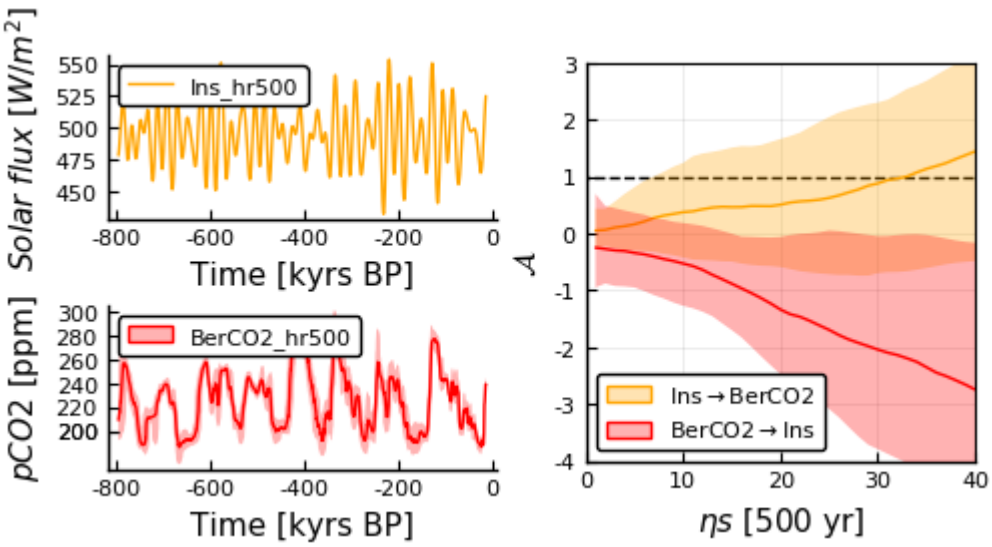
```
# recall the time series on the common grid as X and Y
X = La2004_insol_cut_hr500
Y = B_cut_hr500

# Compute the predictive asymmetries from one time series to another (function d
efined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 0.5, ηmax = Int(20/0.5),
    filepath =  "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_B_hr5
00.jld2" )
```

```
Results are saved in the .jld2 file. 225.574061 seconds (1.39 G allo
cations: 98.129 GiB, 18.86% gc time)
```

In [67]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_B_hr500.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_B_hr500 =
plot(#title = L"$\mathcal{A}$ between pCO2 and insolation",
    xlims = (0, ηmax),
    xticks = (0 : 2*5 : ηmax),
    ylims = (-4,3), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [500 yr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("Ins", L"\rightarrow", "BerCO2"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("BerCO2", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :red
    )


# join plots of time series and pa results to a results subplot
plot_overview_La2004_B_hr500 =
plot(layout = grid(2,1),
    size = (500,400),
    plot_La2004_hr500,
```

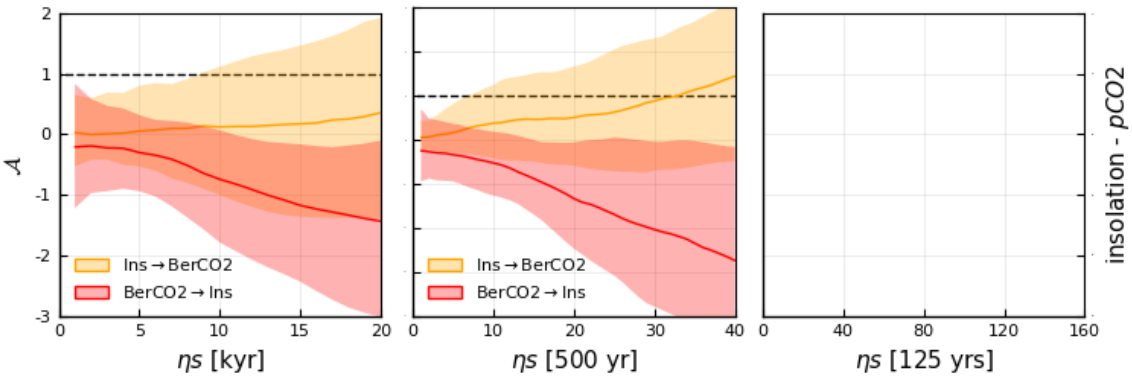
```
    plot_B_hr500
    )

plot_results_La2004_B_hr500 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_La2004_B_hr500,
    plot_normPA_La2004_B_hr500)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/La2004nB_B_hr500.pdf"
)
```



Ensemble plot

In [68]:

```
#### Ins-BerCO2
p4 = plot(plot_normPA_La2004_B, ytickfont = 8, ylabel = string(L"\mathcal{A}"))
p5 = plot(plot_normPA_La2004_B_hr500, ytickfont = false, ylabel = "")
p_emptyB = plot(xlims = (0,160), xticks = (0:5*8:160), xlabel = string(L"ηs", "
 [125 yrs]"),
    ytickfont = false, ymirror = true, ylabel = string("insolation - ", L"pCO
{2}"), border = true)

pe_ins_co2_hr =
plot(p4, p5, p_emptyB,
    layout = grid(1,3),
    size = (750,250),
    bg_legend = :transparent,
    border = true
    )

savefig("../../results_ePalus_ns20/sensitivity_analyses/hr_full_800_ins_co2_hr.p
df")
```



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

La2004 - Lambert

In [76]:

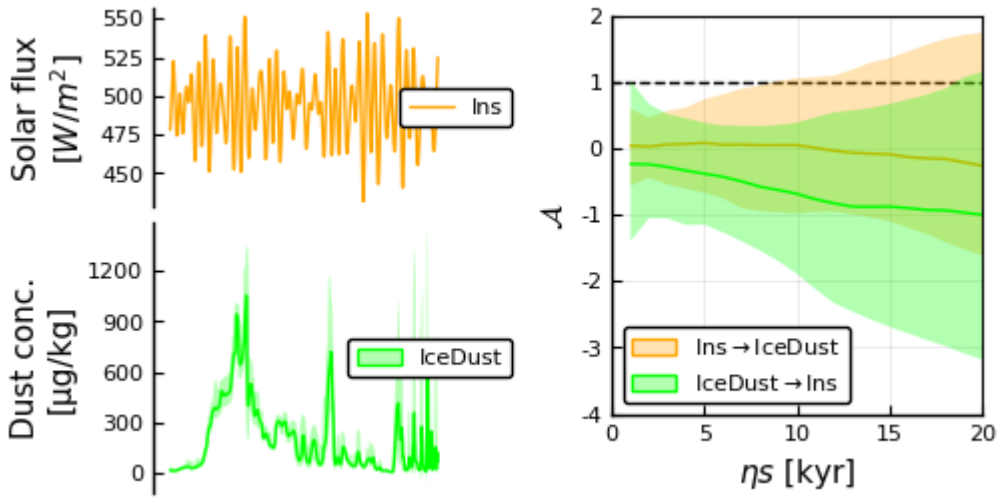
```
#Compute the predictive asymmetry between the two time series


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

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_L.jld
2")
```

```
Results are saved in the .jld2 file.  57.889503 seconds (349.55 M al
locations: 24.947 GiB, 18.82% gc time)
```

In [77]:

```
# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_L.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_L =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-4,2), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel = string(L"ηs"," [kyr]"),
    ylabel = L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("Ins", L"\rightarrow", "IceDust"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :lime)


# join plots of time series and pa results to a results subplot
plot_overview_La2004_L =
plot(layout = grid(2,1),
    size = (500,400),
    plot_La2004,
    plot_L
    )
```

```
plot_results_La2004_L =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_La2004_L,
    plot_normPA_La2004_L)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/La2004nB_L.pdf")
```



- high resolution version (time step of 500 years)

In [93]:

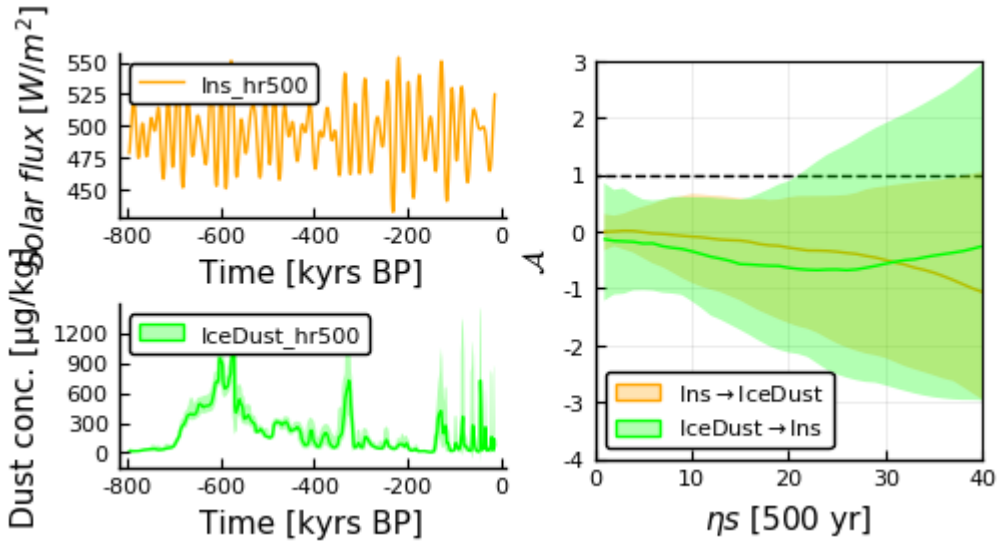
```
# recall the time series on the common grid as X and Y
X = La2004_insol_cut_hr500
Y = L_cut_hr500
binsize_hr500 # recall, time step must be updated before given to the function

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 0.500, ηmax = Int(20/0.500),
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_L_hr50
0.jld2" )
```

```
Results are saved in the .jld2 file. 215.381428 seconds (1.41 G allo
cations: 99.086 GiB, 19.52% gc time)
```

In [94]:

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

# load and normalize the predictive asymmetry results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_L_hr500.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_L_hr500 =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 2*5 : ηmax),
    ylims = (-4,3), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel = string(L"ηs"," [500 yr]"),
    ylabel = L"\mathcal{A}",
    size = (400,400), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("Ins", L"\rightarrow", "IceDust"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("IceDust", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :lime)


# join plots of time series and pa results to a results subplot
plot_overview_La2004_L_hr500 =
plot(layout = grid(2,1),
    size = (500,400),
    plot_La2004_hr500,
    plot_L_hr500
```

```
    )

plot_results_La2004_L_hr500 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_La2004_L_hr500,
    plot_normPA_La2004_L_hr500)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/La2004nB_L_hr500.pdf"
)
```



La2004 - Martinez-García

In [78]:

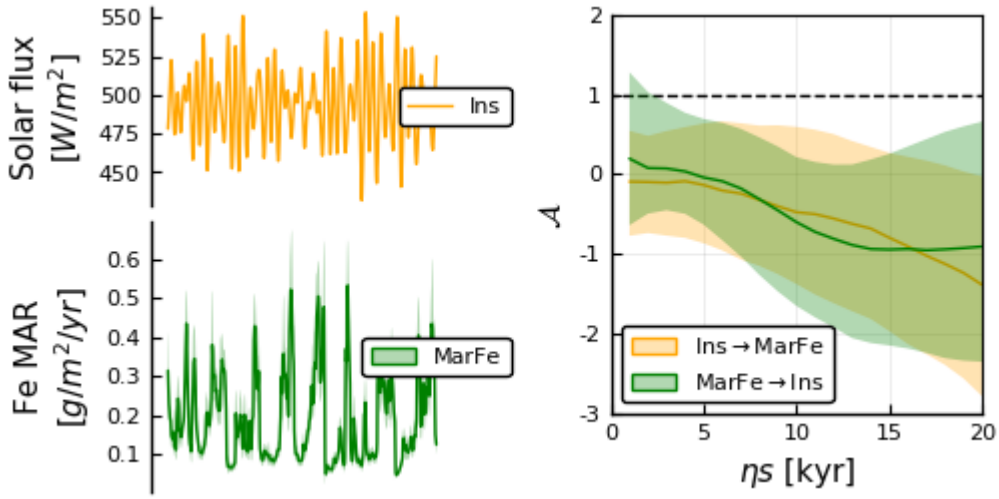
```
# recall the time series on the common grid as X and Y
X = La2004_insol_cut
Y = MG_cut


# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 5,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_MG.jld
2")
```

```
Results are saved in the .jld2 file.  56.198115 seconds (349.42 M al
locations: 24.938 GiB, 19.39% gc time)
```

In [79]:

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

# load and normalize the predictive asymmetry results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_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 = (-3,2), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = 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 = (500,400),
    plot_La2004,
```

```
    plot_MG
    )

plot_results_La2004_MG =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_La2004_MG,
    plot_normPA_La2004_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/La2004nB_MG.pdf")
```



- high resolution version (time step of 500 years)

In [96]:

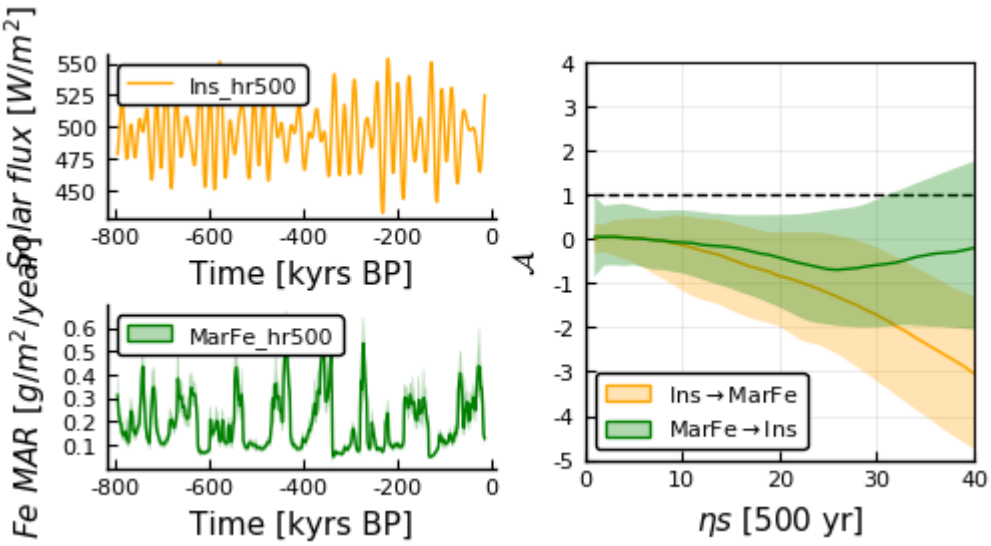
```
#Compute the predictive asymmetry between the two time series


# recall the time series on the common grid as X and Y
X = La2004_insol_cut_hr500 # Northern hemisphere summer insolation
Y = MG_cut_hr500 # Southern Ocean Fe deposition

# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 0.500, ηmax = Int(20/0.500),
    filepath = "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_MG_hr5
00.jld2")
```

```
Results are saved in the .jld2 file. 227.718008 seconds (1.39 G allo
cations: 97.790 GiB, 19.12% gc time)
```

In [99]:

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

# load and normalize the predictive asymmetry results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/full_800/La2004nB_MG_hr500.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_hr500 =
plot(#title = L"$\mathcal{A}$ between insolation and Fe dust"
    xlims = (0, ηmax),
    xticks = (0 : 2*5 : ηmax),
    ylims = (-5,4), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [500 yr]"),
    ylabel =  L"\mathcal{A}",
    size = (400,400), border = true,
    grid = 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_hr500 =
plot(layout = grid(2,1),
    size = (500,400),
    plot_La2004_hr500,
```

```
    plot_MG_hr500
    )

plot_results_La2004_MG_hr500 =
plot(size = (500,250),
    layout = grid(1,2),
    plot_overview_La2004_MG_hr500,
    plot_normPA_La2004_MG_hr500)

savefig("../../results_ePalus_ns20/pa_ResultPlots/full_800/La2004nB_MG_hr500.pdf")
```



Ensemble plot

In [100]:

```
###### Ins-IceDust
p1 = plot(plot_normPA_La2004_L, ytickfont = 8, ylabel = L"\mathcal{A}")
p2 = plot(plot_normPA_La2004_L_hr500, ytickfont = false, ylabel = "")
p3 = plot(plot_normPA_La2004_L_hr125, ytickfont = false, ymirror = true, ylabel
= string("insolation - dust conc."))

pe_ins_dustL_hr =
plot(p1, p2, p3,
    layout = grid(1,3),
    size = (750,250),
    bg_legend = :transparent
    )


##### Ins-MarFe

p5 = plot(plot_normPA_La2004_MG_hr500, ytickfont = false, ylabel = "", xtickfont
= true, xlabel = string(L"ηs", " [500 yrs]"))
p_empty1MG = plot(xlims = (0,160), xticks = (0:5*8:160), xlabel = string(L"ηs",
" [125 yrs]"),
    ytickfont = false, ymirror = true, ylabel = string("insolation - Fe flux"),
border = true)

pe_ins_dustMG_hr =
plot(plot_normPA_La2004_MG, p5, p_empty1MG,
    layout = grid(1,3),
    size = (750,250),
    bg_legend = :transparent,
    border = true
    )

pe_ins_dust_hr =
plot(pe_ins_dustL_hr, pe_ins_dustMG_hr,
    ylims = (-17,6), yticks = (-20:5:20),
    layout = grid(2,1), size = (750,500))

savefig("../../results_ePalus_ns20/sensitivity_analyses/hr_full_800_ins_dust_hr.
pdf")
```

```
UndefVarError: plot_normPA_La2004_L_hr125 not defined

Stacktrace:
 [1] top-level scope at In[100]:4
```

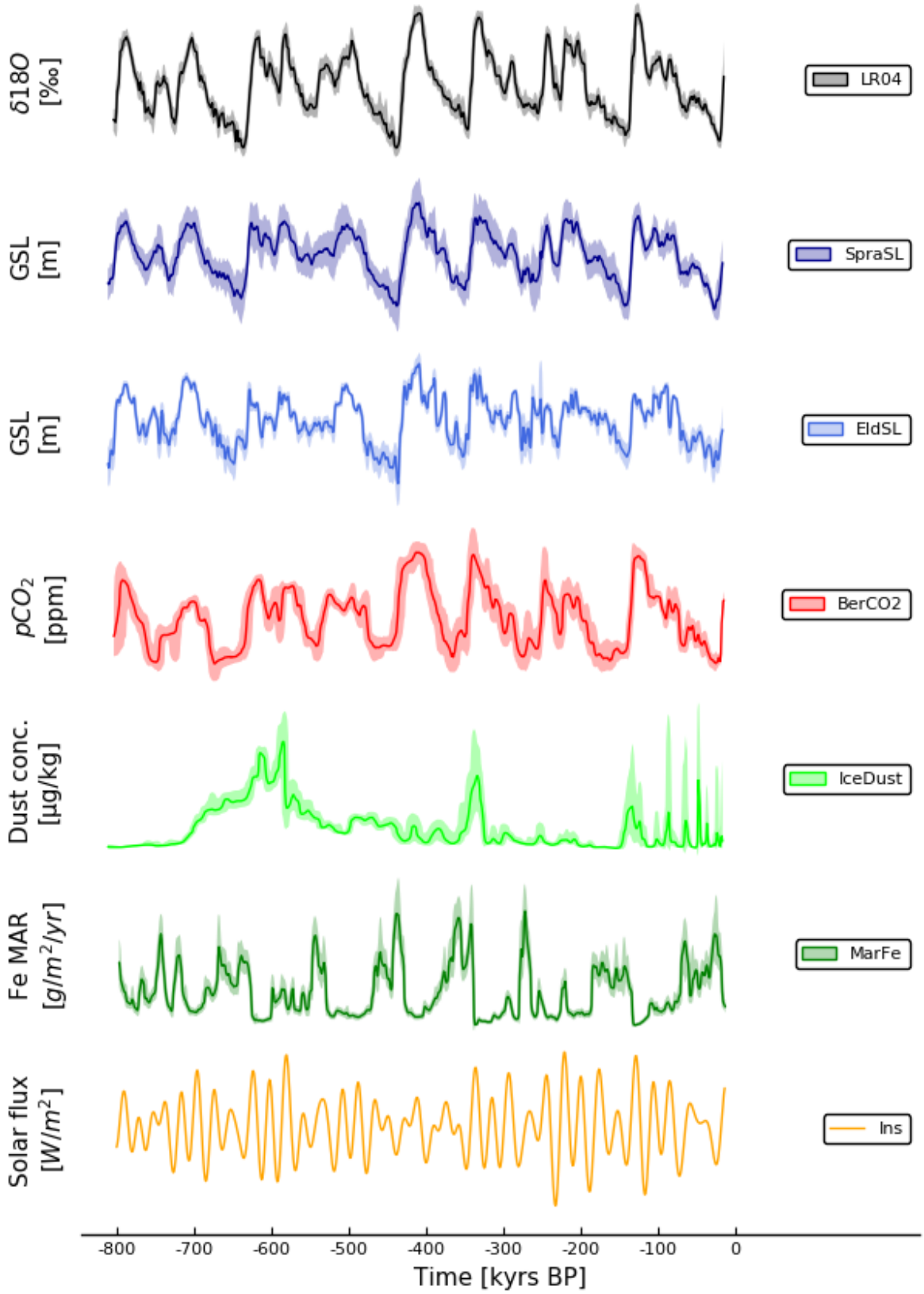
Ensemble plots of results

Overview of time series

In [101]:

```
plot(po_alltimeseries, size = A4, yaxis = :off, ytickfont = :off)
```

Out[101]:



Define some useful plot objects

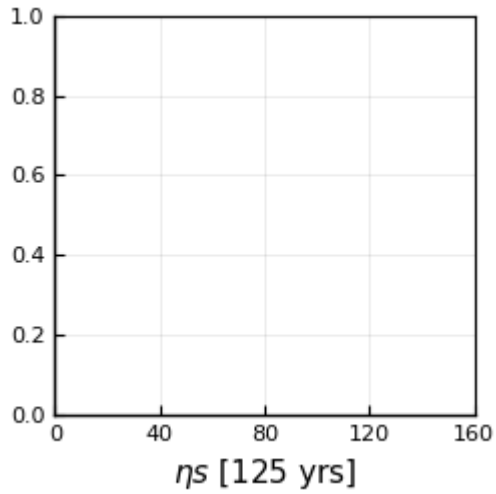
In [ ]:

```
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 [118]:

```
# useful plot objects

p_empty1 = plot(size = (250,250), border = false,  xlabel = string(L"ηs", " [125
yrs]"), xticks = (0:5*8:20*8))
p_empty2 = plot(size = (250,250), border = false,  xlabel = string(L"ηs", " [125
yrs]"), tickfont = false)

p_empty1 = plot(size = (250,250), border = false,  xlabel = string(L"ηs", " [125
yrs]"), xticks = (0:5*8:20*8), xlims = (0,20*8), xtickfont = 8)
```

Out[118]:



Ensemble plot of all results

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

Overview of notebook for Appendices

plot time series and results in as subplots of one large overview plot. Basically summarizes the whole notebook.

Ensemble plot of 1 kyr grid analyses

- Reset ensemble plots for 1 kyr grid (if you've been jumping around the code messing it up)

In [162]:

```
#### 2. **GSL/$\delta^{18}O$ - insolation**


p20 = plot(plot_normPA_LR04_La2004, ylabel = L"\mathcal{A}", xlabel = "", xtickf
ont = false)
p21 = plot(plot_normPA_E_La2004, ylabel = "", ytickfont = false,)
p22 = plot(plot_normPA_R_La2004, ylabel = "", ytickfont = false,)
p23 = plot(plot_normPA_SL_La2004, ylabel = string("GSL - insolation"), ymirror =
true, ytickfont = false)

pe_gsl_insol = plot(
    p20,p21,p23,
    xtickfont = 8,
    layout = grid(1,3), size = (750,250), # without Rohling
    ylims= (-4,3), yticks = (-5:1:5),
    border = true, legend = :bottomleft, bg_legend = :transparent
    )


#### 3. **GSL - pCO2** (post-MPT)


p30 = plot( plot_normPA_LR04_B, ylabel = L"\mathcal{A}")
p31 = plot(plot_normPA_E_B, ylabel = "", ytickfont = false)
p32 = plot(plot_normPA_SL_B, ylabel = "", ytickfont = false)
p33 = plot(plot_normPA_R_B, ylabel = string("GSL - ", L"pCO$_{2}$"), ymirror = t
rue, ytickfont = false)

pe_gsl_co2 = plot(
    #p30, p31,p32,p33,p34,layout = grid(1,5),size = (1250,250),
    p30, p31,p33, layout = grid(1,3),size = (750,250),# without Rohling
    border = true,
    legend = :bottomleft,
    bg_legend = :transparent,
    ylims = (-4,4), yticks = (-5:1:5) )


#### 4. **GSL - dust**

# GSL - IceDust (L)
p0 = plot(plot_normPA_LR04_L, ylabel = L"\mathcal{A}")
p1 = plot(plot_normPA_E_L, ylabel = "", ytickfont = false)
p2 = plot(plot_normPA_SL_L,ylabel = string("GSL - dust"), ymirror = true, ytickf
ont = false)
p3 = plot(plot_normPA_R_L, ylabel = string("GSL - dust"), ymirror = true, ytickf
ont = false)

pe_gsl_dustL = plot(
    p0, p1, p2, layout = grid(1,3), size = (1000,250), #without Rohling
    border = true,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-5,5), yticks = (-10:1:10)
    )


# GSL - MarFe (MG)
p10 = plot(plot_normPA_LR04_MG, ylabel = L"\mathcal{A}")
p11 = plot(plot_normPA_E_MG, ylabel = "", ytickfont = false)
p12 = plot(plot_normPA_SL_MG,ylabel = string("GSL - dust"), ymirror = true, ytic
```

```
kfont = false)
p13 = plot(plot_normPA_R_MG, ylabel = string("GSL - dust"), ymirror = true, ytic
kfont = false)

pe_gsl_dustMG = plot(
    #p10, p11,p12,p13,p14,layout = grid(1,5), size = (1250,250),
    p10, p11,p12,layout = grid(1,3), size = (1000,250), # without Rohling
    border = true, legend = :bottomleft, bg_legend = :transparent,
    ylims = (-8,8), yticks = (-10:1:10))


######## 5. **pCO2 - dust**


p51 = plot(plot_normPA_B_L_edc, xlabel = string(L"ηs", " [kyr]"), xtickfont = 8,
ylabel = string(L"$pCO_{2}$", " - dust
", L"\mathcal{A}"))
p55 = plot(plot_normPA_B_MG, ylabel = "", ytickfont = false)

pe_co2_dust = plot(
    p51, # Bereiter - Lambert
    p55, # Bereiter - Martinez-García
    layout = grid(1,2), size = (500,250),
    border = true,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-7,5), yticks = (-20:1:20)
    )


####### 6. **pCO2 - insolation**

p61 = plot(plot_normPA_La2004_B, bg_legend = :transparent, ylabel = string(L"pCO
$_{2}$", " - insolation
", L"\mathcal{A}"))


##### 7. **insolation - dust**

p71 = plot(plot_normPA_La2004_MG, ymirror = false, ylabel = string("insolation -
dust
", L"\mathcal{A}"))
p76 = plot(plot_normPA_La2004_L, ylabel = "", ytickfont = false)


pe_insol_dust = plot(
    p71,p76,

    layout = grid(1,2),size = (500,250),
    border = true,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-4,2), yticks = (-20:1:20)
    )

;
```

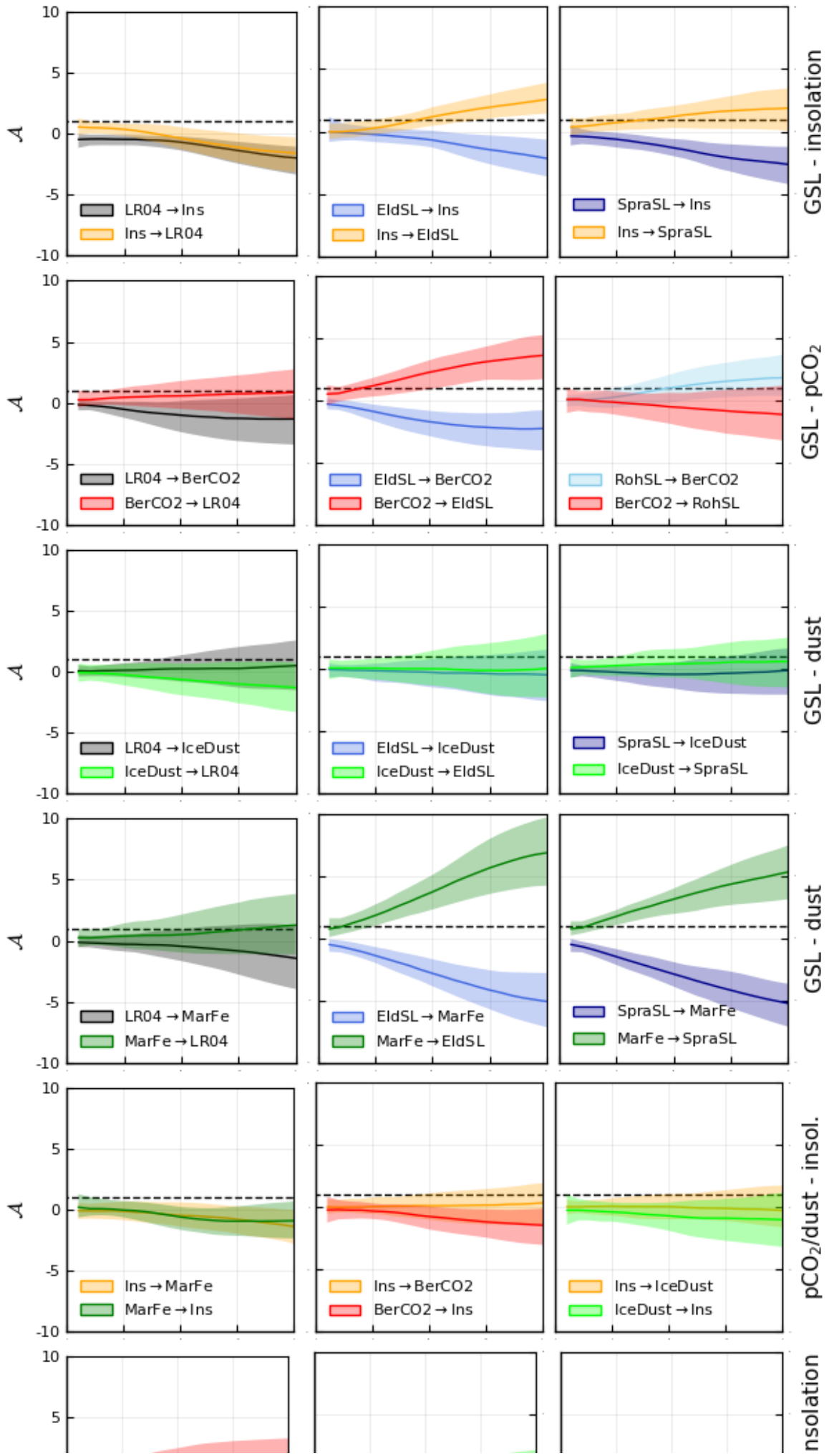
- Overview plot of all results with η = 1 kyr

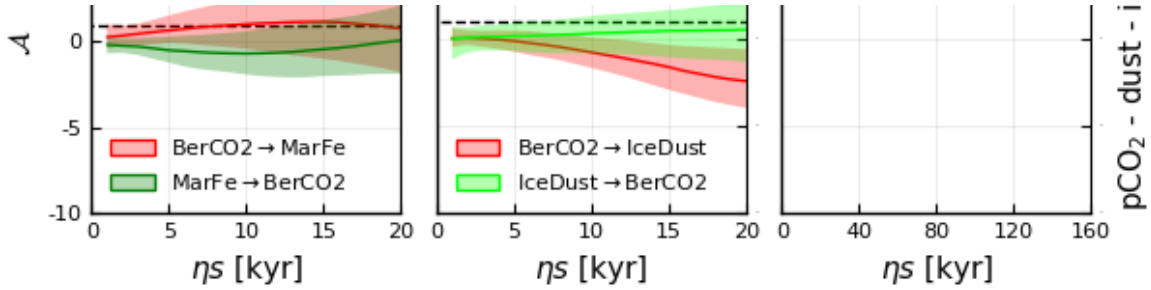
In [174]:

```
# Ensemble plot of 1 kyr grid analyses


# first 4 rows
p91 = plot(pe_gsl_insol, xlabel = "", xlabelticks = "", xtickfont = false)
p92 = plot(pe_gsl_co2, xlabel = "", xtickfont = false)
p93 = plot(pe_gsl_dustL, xlabel = "", xtickfont = false)
p94 = plot(pe_gsl_dustMG, xlabel = "", xtickfont = false)
    #pe_co2_dust,
    #pe_insol_co2,
    #pe_insol_dust,


# fit 3 plots of insolation-dust-CO2 for the bottom row

# fit 3 plots of insolation-dust-CO2 for the bottom row
p81 = plot(plot_normPA_La2004_MG, ylabel = L"\mathcal{A}",ytickfont = 8, ymirror
= false, xlabel = "", xtickfont = false)
p82 = plot(plot_normPA_La2004_B, ylabel = "", ytickfont = false, xlabel = "", xt
ickfont = false)
p83 = plot(plot_normPA_La2004_L,  ytickfont = false, xlabel = "", xtickfont = fa
lse, ylabel = string(L"pCO$_{2}$", "/dust - insol."), ymirror = true)

fit3_insol_co2dust = plot(
    p81, p82, p83,
    layout = grid(1,3), size = (750,250),
    ylims = (-7,5), yticks = (-10:1:10))


# fit 3 plots of dust-CO2 for the bottom row

p84 = plot(plot_normPA_B_MG, ylabel = L"\mathcal{A}", ytickfont = 8, xlabel = st
ring(L"ηs", " [kyr]"), xtickfont = 8)
p85 = plot(plot_normPA_B_L_edc, ytickfont = false, xlabel = string(L"ηs", " [ky
r]"), xtickfont = 8, ylabel = "")
p86 = plot(p_empty1, ylabel = string(L"pCO$_{2}$", " - dust - insolation"), ymir
ror = true, ytickfont = false, xlabel = string(L"ηs", " [kyr]"), xtickfont = 8)

fit3_co2_dust = plot(
    p84,p85,p86,
    layout = grid(1,3), size = (750,250),
    ylims = (-7,5), yticks = (-10:1:10))


A4 = (72*8.27, 72*11.79) # A4 dimensions
A3 = (72*8.27*2, 72*11.79*2) # A3 dimensions

####
pe_allresults =
plot(p91, p92, p93, p94, fit3_insol_co2dust, fit3_co2_dust,
    layout = grid(6,1),
    size = (3*200, 6*200),
    bg_legend = :transparent,
    ylims = (-10,10), yticks = (-10:5:10)
    )

# for i in 1:25
#    annotate!(2,7, panel_labels[i], subplot = i)
# end
```

```
pe_allresults
#savefig("../../results_ePalus_ns20/e_allresults/MA__full_800_wPL_square.pdf")
#savefig("../../figurar/Final_Results/full_800/pe_allresults_1kyrgrid_wPL_squar
e.pdf")
```

Out[174]:

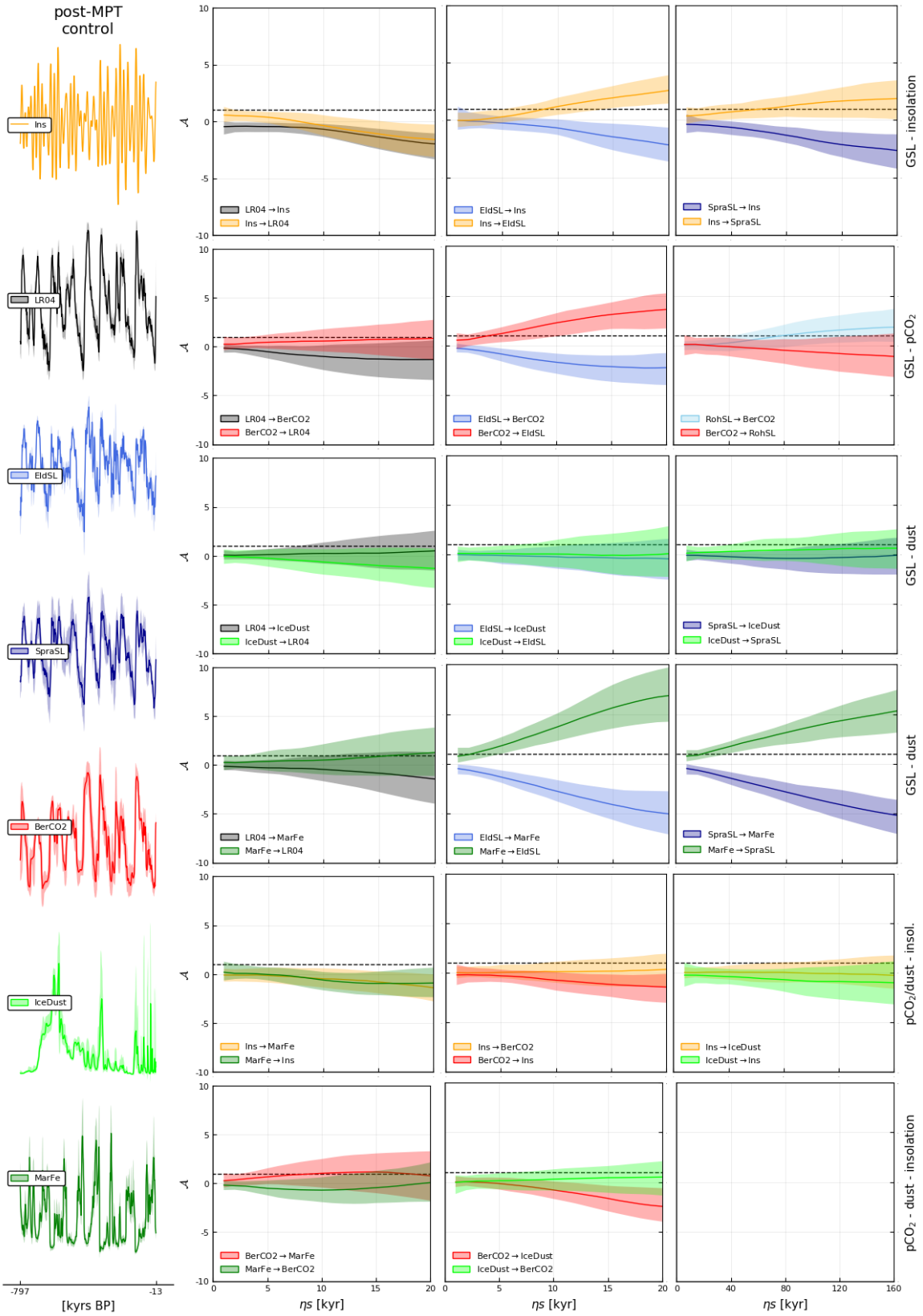

10
5
0
-5
-10
$\mathcal{A}$
LR04 → Ins
Ins → LR04
EIdSL → Ins
Ins → EIdSL
SpraSL → Ins
Ins → SpraSL
GSL - insolation
LR04 → BerCO2
BerCO2 → LR04
EIdSL → BerCO2
BerCO2 → EIdSL
RohSL → BerCO2
BerCO2 → RohSL
GSL - pCO_2
LR04 → IceDust
IceDust → LR04
EIdSL → IceDust
IceDust → EIdSL
SpraSL → IceDust
IceDust → SpraSL
GSL - dust
LR04 → MarFe
MarFe → LR04
EIdSL → MarFe
MarFe → EIdSL
SpraSL → MarFe
MarFe → SpraSL
GSL - dust
Ins → MarFe
MarFe → Ins
Ins → BerCO2
BerCO2 → Ins
Ins → IceDust
IceDust → Ins
pCO_2/dust - insol.
nsolation



- Join overview results plot with the time series plot (- figure to include in thesis)

In [180]:

```
# overview time series

po0 = plot(plot_La2004, xlabel = "", xaxis = :off, title = "post-MPT
control")
#    (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) # Rohling results excluded due to
 sapropelic intervals
po5 = plot(plot_SL, xlabel = "", xaxis = :off)
#po6 = plot(plot_G, xlabel = "", xaxis = :off)
po7 = plot(plot_B, xlabel = "", xaxis = :off)
po8 = plot(plot_L, xlabel = "", xaxis = :off)
po9 = plot(plot_MG, xlabel = "", xaxis = :off)
p_xaxis = plot(xlabel = "[kyrs BP]", xaxis = :on, xmirror = false, yaxis = false
, ylabel = false,grid = false)

l1 = @layout [a;b;c;d;e;f;g;h{0.01h}] #defines layout
po_alltimeseries = plot(po0, po1, po3, po5, po7, po8, po9, p_xaxis,
    layout = l1, #grid(7,1),
    size = (72*8.27, 72*11.79), # A4,
    xlims = (gridstart-100, gridend+100), xticks = (gridstart : gridend-gridstar
t  : gridend),
    ylabel = "", ytickfont = false, yaxis = :off,
    legend = :left, bg_legend = :white, font_legend = 8)

l2 = @layout [a{0.2w} b{0.8w}] # defines layout
P_Appendix = plot(po_alltimeseries, pe_allresults, layout = l2, size = A3)
#savefig("../../results_ePalus_ns20/e_allresults/MA__full_800_AppendixA3.pdf")
#savefig("../../figurar/Final_Results/MA__full_800_AppendixA3.pdf")
```


post-MPT
control
Ins
LR04
EldSL
SpraSL
BerCO2
IceDust
MarFe
-797
-13
[kyrs BP]
$\mathcal{A}$
LR04 → Ins
Ins → LR04
EldSL → Ins
Ins → EldSL
SpraSL → Ins
Ins → SpraSL
LR04 → BerCO2
BerCO2 → LR04
EldSL → BerCO2
BerCO2 → EldSL
RohSL → BerCO2
BerCO2 → RohSL
LR04 → IceDust
IceDust → LR04
EldSL → IceDust
IceDust → EldSL
SpraSL → IceDust
IceDust → SpraSL
LR04 → MarFe
MarFe → LR04
EldSL → MarFe
MarFe → EldSL
SpraSL → MarFe
MarFe → SpraSL
Ins → MarFe
MarFe → Ins
Ins → BerCO2
BerCO2 → Ins
Ins → IceDust
IceDust → Ins
BerCO2 → MarFe
MarFe → BerCO2
BerCO2 → IceDust
IceDust → BerCO2
ηs [kyr]
GSL - insolation
GSL - pCO2
GSL - dust
GSL - dust
pCO2/dust - insol.
pCO2 - dust - insolation

Ensemble analyses η = 1 kyr and η = 500 yrs

In [ ]:

```
# pe_allresults_whr500

# vertical array of hr500-plots, to plot alongside pe_allresults_1kyrgird

p_hr500b = plot(plot_normPA_G_La2004nB_hr500, xlabel = "", xtickfont = false)
p_hr500c = plot(plot_normPA_G_B_hr500,        xlabel = "", xtickfont = false)
p_hr500d = plot(plot_normPA_G_L_hr500,        xlabel = "", xtickfont = false)
p_hr500e = plot(plot_normPA_G_MG_hr500,       xlabel = "", xtickfont = false)
p_hr500f = plot(plot_normPA_B_L_edc_hr500, xlabel = string(L"ηs", " [125 yrs]"),
xtickfont = 8)

pe_hr500_vArray5 = plot(p_hr500b, p_hr500c, p_hr500d, p_hr500e, p_hr500f,
    layout = grid(5,1), size = (150,5*150),
    ylabel = "", ytickfont = false)

##### plot pe_allresults, including findings from hr500

l = @layout [a b{0.2w}]
pe_allresults_whr = plot(pe_allresults, pe_hr500_vArray5, layout = l, ylims = (-
12,10), yticks = (-20:5:20)
    , size = (5*200,5*200))

savefig("../../results_ePalus_ns20/e_allresults/MA__full_800_wHR_square.pdf")
savefig("../../figurar/Final_Results/full_800/pe_allresults_wHR_square.pdf")
```

In [ ]:

```
l2 = @layout [a{0.2w} b{0.8w}]
P_Appendix_whr = plot(po_alltimeseries, pe_allresults_whr, layout = l2, size = A
3) # A3 size
#savefig("../../results_ePalus_ns20/e_allresults/MA__full_800_whr_AppendixA3.pd
f")
#savefig("../../figurar/Final_Results/full_800/MA__full_800_whr_AppendixA3.pdf")
```

3*3 overview of hr analyses with time step η = 500

In [ ]:

```
# hr500 9*9 plot

p1 = plot(plot_normPA_G_L_hr500, xlabel = "", xtickfont = false, ylabel = L"\mat
hcal{A}", ytickfont = 8)
p2 = plot(plot_normPA_B_L_edc_hr500, xlabel = "", xtickfont = false, ylabel = ""
, ytickfont = false)
p3 = plot(plot_normPA_La2004_L_hr500, xlabel = "", xtickfont = false, ylabel =
"", ytickfont = false)

p4 = plot(plot_normPA_G_MG_hr500, xlabel = "", xtickfont = false, ylabel = L"\ma
thcal{A}", ytickfont = 8)
p5 = plot(plot_normPA_B_MG_hr500, xlabel = "", xtickfont = false, ylabel = "", y
tickfont = false)
p6 = plot(plot_normPA_La2004_MG_hr500, xlabel = "", xtickfont = false, ylabel =
"", ytickfont = false)

p7 = plot(plot_normPA_G_B_hr500, xlabel = string(L"ηs", " [500 yrs]"), ylabel =
L"\mathcal{A}", tickfont = 8)
p8 = plot(plot_normPA_G_La2004nB_hr500, xlabel = string(L"ηs", " [500 yrs]"), yl
abel = "", ytickfont = false, xtickfont = 8)
p9 = plot(plot_normPA_La2004_B_hr500, xlabel = string(L"ηs", " [500 yrs]"), ylab
el = "", ytickfont = false, xtickfont  = 8)

p_hr500 = plot(
    p1,p2,p3,p4,p5,p6,p7,p8,p9,
    layout = grid(3,3), size = (3*250, 3*250), bg_legend = :transparent,
    ylims = (-12,8), yticks = (-20:2:20), border = true,
    )


#savefig("../../results_ePalus_ns20/sensitivity_analyses/MA__hr500_full_800_noP
L.pdf")
#savefig("../../figurar/Final_Results/full_800/pe_allresults_hr500_noPL.pdf")

# for i in 1:9
#    annotate!(4,6, panel_labels[i], subplot = i)
# end

plot(p_hr500)
#savefig("../../results_ePalus_ns20/sensitivity_analyses/MA__hr500_full_800_noP
L.pdf")
#savefig("../../figurar/Final_Results/full_800/pe_allresults_hr500_noPL.pdf")
```

Overview plot of sensitivity analyses on η (hrSA)

- define the hrSA plots again (have been wiped in code above)

In [ ]:

```
###### GraSL-Ins hrSA
pAA_Gins1 = plot(plot_normPA_G_La2004nB, ymirror = false, ylabel = L"\mathcal
{A}", tickfont = 8,  xlabel = string(L"ηs", " [kyr]"))
pAA_Gins2 = plot(plot_normPA_G_La2004nB_hr500, ylabel = "", ytickfont = false,)
pAA_Gins3 = plot(plot_normPA_G_La2004nB_hr125, ylabel = string("GSL - insolatio
n"), ymirror = true, ytickfont = false)

pe_gsl_ins_hr = plot(pAA_Gins1, pAA_Gins2, pAA_Gins3,
    layout =  grid(1,3), size = (750,250),
    border = true, bg_legend = :transparent,
    ylims = (-12,5), yticks = (-20:1:20))

###### GraSL-BerCO2

p35 = plot(plot_normPA_G_B, xlabel = string(L"ηs", " [kyr]"), ylabel = string(L
"\mathcal{A}"), ymirror = false, xtickfont = 8, ytickfont = 8)
p36 = plot(plot_normPA_G_B_hr500, ylabel = "", ytickfont = false)
#p37 = plot(plot_normPA_G_B_hr125, ylabel = "", ytickfont = false) # over-interp
olation, but would be interesting to see
p_empty1 = plot(size = (250,250), border = false,  xlabel = string(L"ηs", " [125
yrs]"), xlims = (0,160), xtick = (0:5*8:20*8), xtickfont = 8,
    ylabel = string("GSL - ", L"pCO{2}"), ymirror = true, ytickfont = false)


pe_gsl_co2_hr = plot(
    p35, p36, p_empty1,# p37,
    layout = grid(1,3), size = (750,250),
    border = true,
    legend = :bottomleft,
    bg_legend = :transparent,
    ylims = (-6,6), yticks = (-10:1:10)
    )

############ GraSL - IceDust

p6 = plot(plot_normPA_G_L, ytickfont = 8, ymirror = false, ylabel = L"\mathcal
{A}")
p7 = plot(plot_normPA_G_L_hr500, ylabel = "", ytickfont = false, xlabel = string
(L"ηs", " [500 yrs]"))
p8 = plot(plot_normPA_G_L_hr125, ytickfont = false, xlabel = string(L"ηs", " [12
5 yrs]"),
ylabel = string("GSL - dust conc."), ymirror = true)

pe_gsl_dustL_hr =
plot(p6, p7, p8,
    layout = grid(1,3), size = (750,250),
    border = true, legend = :bottomleft, bg_legend = :transparent,
    ylims = (-20,15), yticks = (-20:5:20),
    xlabel = "", xtickfont = false)

####### GraSL - MarFe

p16 = plot(plot_normPA_G_MG, ytickfont = 8, ymirror = false, ylabel = string(L
"\mathcal{A}"))
p17 = plot(plot_normPA_G_MG_hr500, ylabel = "", ytickfont = false, xlabel = stri
ng(L"ηs", " [500 yrs]"), )
p_empty1 = plot(xlims = (0,160), xticks = (0:5*8:160), ytickfont = false, xlabel
= string(L"ηs", " [125 yrs]"),
    ymirror = true, ylabel = string("GSL - Fe flux"))
```

```
pe_gsl_dustMG_hr =
plot(p16, p17, p_empty1,
    layout = grid(1,3), size = (750,250),
    border = true, legend = :bottomleft, bg_legend = :transparent,
    ylims = (-20,15), yticks = (-20:5:20),
    xtickfont = 8)

#### BerCO2-IceDust

p51 = plot(plot_normPA_B_L_edc, ylabel = L"\mathcal{A}", ymirror = false, ytickf
ont = 8)
p52 = plot(plot_normPA_B_L_edc_hr500, ylabel = "", ytickfont = false)
p_empty1 = plot(xlims = (0,160), xticks = (0:5*8:160), ytickfont = false,
    xlabel = string(L"ηs", " [125 yrs]"),
    ymirror = true, ylabel = string(L"pCO_{2}", " - dust conc."))

pe_co2_dustL_hr = plot(p51, p52,  p_empty1, # Bereiter - Lambert
    layout = grid(1,3), xlabel = "", xtickfont = false)

#### BerCO2-MarFe

p53 = plot(plot_normPA_B_MG, ytickfont = 8, ylabel = L"\mathcal{A}", ymirror = f
alse)
p54 = plot(plot_normPA_B_MG_hr500, ylabel = "", ytickfont = false)
p_empty2 = plot(xlims = (0,160), xticks = (0:5*8:160), ytickfont = false,
    xlabel = string(L"ηs", " [125 yrs]"),
    ymirror = true, ylabel = string(L"pCO_{2}", " - Fe flux")
    )

pe_co2_dustMG_hr = plot( p53, p54,  p_empty2, # Bereiter - Martinez-García
    layout = grid(1,3),)

#### Ins-BerCO2
p_BI1 = plot(plot_normPA_La2004_B, ylabel = L"\mathcal{A}", ytickfont = 8)
p_BI5 = plot(plot_normPA_La2004_B_hr500, ytickfont = false, ylabel = "", )
p_emptyBI = plot(xlims = (0,160), xticks = (0:5*8:160), xlabel = string(L"ηs", "
[125 yrs]"),
    ytickfont = false, ymirror = true, ylabel = string("insolation - ", L"pCO
{2}"), border = true)

pe_ins_co2_hr =
plot(p_BI1, p_BI5, p_emptyBI,
    layout = grid(1,3),
    size = (750,250),
    bg_legend = :transparent,
    border = true
    )


###### Ins-IceDust
p_IL1 = plot(plot_normPA_La2004_L, ylabel = L"\mathcal{A}", ytickfont = 8)
p_IL2 = plot(plot_normPA_La2004_L_hr500, ytickfont = false, ylabel = "")
p_IL3 = plot(plot_normPA_La2004_L_hr125, ytickfont = false, ymirror = true, ylab
el = string("insolation - dust conc."))

pe_ins_dustL_hr =
plot(p_IL1,p_IL2, p_IL3,
    layout = grid(1,3),
    size = (750,250),
    bg_legend = :transparent
```

```
    )


##### Ins-MarFe

p5 = plot(plot_normPA_La2004_MG_hr500, ytickfont = false, ylabel = "", xtickfont
= 8, xlabel = string(L"ηs", " [500 yrs]"))
p_empty1MG = plot(xlims = (0,160), xticks = (0:5*8:160), xlabel = string(L"ηs",
" [125 yrs]"),
    ytickfont = false, ymirror = true, ylabel = string("insolation - Fe flux"),
border = true)

pe_ins_dustMG_hr =
plot(plot_normPA_La2004_MG, p5, p_empty1MG,
    layout = grid(1,3),
    size = (750,250),
    bg_legend = :transparent,
    border = true
    )
;
```

- define the hrSA plot

In [ ]:

```
#sensitivity analyses on eta

p_hr_a = plot(pe_gsl_ins_hr, xlabel = "", xtickfont = false)
p_hr_b = plot(pe_gsl_co2_hr, xlabel = "", xtickfont = false)
p_hr_c = plot(pe_gsl_dustL_hr,        xlabel = "", xtickfont = false)
p_hr_d = plot(pe_gsl_dustMG_hr,       xlabel = "", xtickfont = false)
p_hr_e = plot(pe_co2_dustL_hr, xlabel = "", xtickfont = false)
p_hr_f = plot(pe_co2_dustMG_hr, xlabel = "", xtickfont = false)
p_hr_g = plot(pe_ins_co2_hr, xlabel = "", xtickfont = false)
p_hr_h = plot(pe_ins_dustL_hr, xlabel = "", xtickfont = false)
p_hr_i = plot(pe_ins_dustMG_hr)


pe_hr_all =
plot(p_hr_a, p_hr_b, p_hr_c, p_hr_d, p_hr_e, p_hr_f, p_hr_g, p_hr_h, p_hr_i,
    layout = grid(9,1),
    border = true,
    ylims = (-20,15), yticks = (-20:5:20),
    #plot_title = "Sensitivity analysis of time step (η)", # Want to make a figu
re title, but kwarg discontinued
    size = (600, 9*200),
    #size = (72*8.27*, 72*11.79*) # A3 size
    )

#savefig("../../results_ePalus_ns20/sensitivity_analyses/hr_full_800_hrALL_noPan
ellabels.pdf")

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)

#for i in 1:27
#    annotate!(2,12, panel_labels[i], subplot = i)
#end

plot(pe_hr_all)
#savefig("../../results_ePalus_ns20/sensitivity_analyses/hrSA_full_800_hrALL_noP
L.pdf")
#savefig("../../figurar/Final_Results/full_800/hrSA_wPL.pdf")
```