time grid label for this notebook: `NBR_synMPT_Cgrid_e4_La2004nB`

Predictive asymmetry analysis for climate system dynamics during the MPT.

Written by Maria Salem, in supplement to master thesis.

In this notebook we compute the normalized predictive asymmetry ($\mathcal{A}$) between empirical climate proxy records over the **time window 1240-1090 ka BP**. The aim is to deduce causal coupling strength and directionality in the climate system between ice volume, northern hemisphere summer insolation, atmospheric pCO2 concentration, and Southern Ocean Fe fertilization prior to the Mid-Pleistocene Transition.

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

In [2]:

```
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 grid
```

```
SystemError: truncate: No space left on device

Stacktrace:
 [1] #systemerror#44(::Nothing, ::typeof(systemerror), ::String, ::B
ool) at ./error.jl:134
 [2] truncate at ./error.jl:134 [inlined]
 [3] grow at /Users/maria/.julia/packages/JLD2/hB4ya/src/mmapio.jl:1
30 [inlined]
 [4] resize!(::JLD2.MmapIO, ::Ptr{Nothing}) at /Users/maria/.julia/p
ackages/JLD2/hB4ya/src/mmapio.jl:140
 [5] save_group(::JLD2.Group{JLD2.JLDFile{JLD2.MmapIO}}) at /Users/m
aria/.julia/packages/JLD2/hB4ya/src/mmapio.jl:239
 [6] close(::JLD2.JLDFile{JLD2.MmapIO}) at /Users/maria/.julia/packa
ges/JLD2/hB4ya/src/JLD2.jl:367
 [7] #jldopen#31(::Base.Iterators.Pairs{Union{},Union{},Tuple{},Name
dTuple{(),Tuple{}}}, ::typeof(jldopen), ::getfield(Main, Symbol("##5
6#57")){Int64,UncertainIndexValueDataset{UncertainIndexDataset,Uncer
tainValueDataset},UncertainIndexValueDataset{UncertainIndexDataset,U
ncertainValueDataset},Array{Array{Float64,1},1},Array{Array{Float64,
1},1},Array{Array{Float64,1},1},Array{Array{Float64,1},1}}, ::Strin
g, ::Vararg{String,N} where N) at /Users/maria/.julia/packages/JLD2/
hB4ya/src/loadsave.jl:6
 [8] jldopen at /Users/maria/.julia/packages/JLD2/hB4ya/src/loadsav
e.jl:2 [inlined]
 [9] macro expansion at /Users/maria/.julia/packages/JLD2/hB4ya/src/
loadsave.jl:66 [inlined]
 [10] #computePredictiveAsymmetries#55(::Int64, ::Int64, ::Int64, ::
Int64, ::RandomSequences, ::String, ::typeof(computePredictiveAsymme
tries), ::UncertainIndexValueDataset{UncertainIndexDataset,Uncertain
ValueDataset}, ::UncertainIndexValueDataset{UncertainIndexDataset,Un
certainValueDataset}) at /Users/maria/Jottacloud/MASTER_2.0/Koding/N
BRs_ePalus_ns20/NB3_Toolbox_ns20.ipynb:In[+9]:47
 [11] (::getfield(Main, Symbol("#kw##computePredictiveAsymmetrie
s")))(::NamedTuple{(:timestep, :ηmax, :filepath),Tuple{Int64,Int64,St
ring}}, ::typeof(computePredictiveAsymmetries), ::UncertainIndexValu
eDataset{UncertainIndexDataset,UncertainValueDataset}, ::UncertainIn
dexValueDataset{UncertainIndexDataset,UncertainValueDataset}) at ./n
one:0
 [12] top-level scope at util.jl:156
 [13] include_string(::Module, ::String, ::String) at ./loading.jl:1
064
 [14] my_include_string(::Module, ::String, ::String, ::Nothing, ::B
ool) at /Users/maria/.julia/packages/NBInclude/mNhzW/src/NBInclude.j
l:29
 [15] #nbinclude#1(::Bool, ::UnitRange{Int64}, ::Regex, ::typeof(ide
ntity), ::Bool, ::typeof(nbinclude), ::Module, ::String) at /Users/m
aria/.julia/packages/NBInclude/mNhzW/src/NBInclude.jl:82
 [16] nbinclude(::Module, ::String) at /Users/maria/.julia/packages/
NBInclude/mNhzW/src/NBInclude.jl:53
 [17] top-level scope at In[2]:2
```

The following records are used for analysis (references used as record labels, in bold):

- **Ins** / La2004: Numerical solution for northern hemisphere insolation (top of atmosphere solar flux mean for summer solstice at $65^{\circ} N$), by *Laskar et al. (2004)*.
- **LR04**: $\delta^{18}O$ global reference stack - a principal component analysis of 57 $\delta^{18}O$ records, by *Lisiecki & Raymo (2005)*.
- **EldSL** / E: Temperature deconvolution of $\delta^{18}O$ signal in marine sediment core from the Chatnam Rise (Pacific Ocean), spanning the last 1.5 Myrs, by *Elderfield et al. (2012)*.
- **RohSL** / R: Relative sea level at Straight of Gibraltar, spanning the last 5.3 Myrs (Rohling et al., 2014).
- **ChaCO2** / C: High resolution $\delta^{11}B$ proxy record for pCO2, spanning a 150 kyr interval of the MPT, by *Chalk et al. (2017)*.
- **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:

Get the time series on a common time grid

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

Run pairwise analyses for predictive asymmetry between the time series

1. Compute the normalized predictive asymmetry (function defined in NB3) for each time series pair, and save the results in a .jld2 file.
2. Produce results plot of the normalized predictive asymmetry for each time series pair. **Produce an overview plot of predictive asymmetry results over the notebook's time interval**
3. Produce an overview plot of the ensemble of predictive asymmetry results for the time period here studied. The ensemble plot will be use in the results chapter of main text.

Get the time series on a common time grid

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

1. Decide on common time grid

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

In [1]:

```
# 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_E,
        tmin_R,
        tmin_C, # the Chalk record is the shortest record and delimits the time
 interval analysed.
        tmin_MG
        ])

# the record with the "earliest end" the end of the common grid
gridend = minimum([
        tmax_LR04,
        tmax_La2004,
        tmax_E,
        tmax_R,
        tmax_C,
        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.
```

```
UndefVarError: tmin_LR04 not defined

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

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 [4]:

```
# Recall the binsize
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 on
a hr grid, 125 year timestep (Grant, Chalk, La2004)
binsize_hr500 = 0.5     # Martinez-García and La2004 are additionally binned on a
grid with a 500 year timestep

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 `synMPT_Cgrid_e4_La2004nB` .

In [73]:

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

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

# Additional high resolution grids
commongrid_hr125 = gridstart : 0.125 : gridend
Binmidpoints_commongrid_hr125 = [commongrid_hr125[i] for i in 1:length(commongri
d_hr125)]

commongrid_hr500 = gridstart : 0.500 : gridend
Binmidpoints_commongrid_hr500 = [commongrid_hr500[i] for i in 1:length(commongri
d_hr500)]

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

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

```
-1240.0:1.0:-1092.0 defines the main common grid for analyses in thi
s notebook.
```

Time series length.

In [6]:

```
# time series lengths in this notebook

N_1 = length(Binmidpoints_commongrid)             # 1kyr time step
N_hr500 = length(Binmidpoints_commongrid_hr500) # 500 yr time step
N_hr125 = length(Binmidpoints_commongrid_hr125) # 125 yr timestep


print("Time series on the 1 kyr grid are of length ", N_1, " datapoints.
Higher resolution time series  with 500 yr and 125 time step grids are of length
", N_hr500, " and ", N_hr125," datapoints, respectively.")
```

```
Time series on the 1 kyr grid are of length 149 datapoints.
Higher resolution time series  with 500 yr and 125 time step grids a
re of length 297 and 1185 datapoints, respectively.
```

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

In [7]:

```
# time series lengths in this notebook
#print("Time series on the 1 kyr grid are of length ", length(Binmidpoints_commo
ngrid), " - we will use f = 1.5 in interpretation of the results for the 1 kyr t
imestep grid.
#High resolution analyses with 500 yr time step grid are of length ", length(Bin
midpoints_commongrid_hr500), " - we will use f = 1 in interpretation of the resu
lts.
#High resolution analyses with 125 yr time step are of length ", length(Binmidpo
ints_commongrid_hr125), " - we will use f = 1 in interpretation of the results.
 ")

f = 1.5
```

Out[7]:

```
1.5
```

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

In [8]:

```
# Palus horizon
function PalusHorizon(
    # input arguments
        N::Number; # time series length (number of datapoints)
        eD::Int = 3 # embedding dimension in the VisitationFrequencyTest - we ha
ve used the default 3
        )

    # method
    ϵ = Int(round(N^(1/(eD+1))))
    return ϵ_Palus
end
```

Out[8]:

```
PalusHorizon (generic function with 1 method)
```

In [9]:

```
# Palus horizon
eD = 3#  embedding dimension in the VisitationFrequencyTest - we have used the d
efault 3

# 1kyr time step
ϵ = Int(round( N_1^(1/(eD+1)) ))
print("The Palus horizon of ϵ is
    ", ϵ, " for the 1 kyr timestep grid")


# 500 yr time step
ϵ = Int(round( N_hr500^(1/(eD+1)) ))
print("
    ", ϵ, " for the 500 yr timestep grid")


# 125 yr timestep
ϵ = Int(round( N_hr125^(1/(eD+1)) ))
print("
and ", ϵ, " for the 125 yr timestep grid")
```

```
The Palus horizon of ϵ is
    3 for the 1 kyr timestep grid
    4 for the 500 yr timestep grid
and 6 for the 125 yr timestep grid
```

We use the Palus horizon to determine state space binning for the high resolution analyses, but we do not go on a coarser state space grid than ϵ = 4 (see sensitivity analyson ϵ in appendices for reasoning). This means, in this notebook we change the default argument to ϵ = 4 for analyses on the 1 kyr grid.

The Palus approximation of optimal binning (default definition of ϵ) is 3 for time series of this length. This ϵ yields a coarse grained state space grid with only 3^3 = 27 bins. According to our sensitivity analyses (see appendices), this is insufficient to give robust estimations of transfer entropies. We therefore set as a minimum ϵ = 4, meaning 4^4 = 64 bins is the coarsest gridding of state space allowed for estimating transfer entropy.

In [10]:

```
# Default epsilon (Palus horizon)
N = length(Binmidpoints_commongrid)
eD = 3 #embedding dimension
ϵ_Palus = Int(round(N^(1/(eD+1)) )) # Epsilon defined inside the computePredicti
veAsymmetry function. By default, ϵ is defined according to the Palus horizon

print("The Palus approximation of optimal binning (default definition of ϵ) is "
, ϵ_Palus, " for time series of this length. This ϵ yields a coarse grained state
space grid with only ", ϵ_Palus^(eD), " bins.
According to our sensitivity analyes, this is insufficient to give robust estima
tions of transfer entropies.
We therefore set as a minimum ϵ = 4, meaning ", 4^(eD), " bins is the coarsest g
ridding of state space allowed for estimating transfer entropy.")
```

```
The Palus approximation of optimal binning (default definition of ϵ)
is 3 for time series of this length. This ϵ yields a coarse grained s
tate space grid with only 27 bins.
According to our sensitivity analyes, this is insufficient to give r
obust estimations of transfer entropies.
We therefore set as a minimum ϵ = 4, meaning 64 bins is the coarsest
gridding of state space allowed for estimating transfer entropy.
```

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

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

Insolation - La2004

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

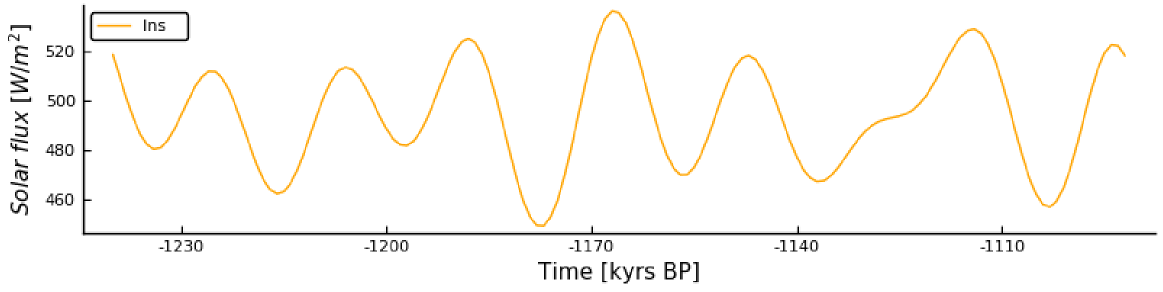
- Default version (`La2004_insol_cut`) is on a grid with 1 kyr timestep between each insolation value (as in the original dataset).

In [77]:

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

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


# define the time series plot

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

```
-1240.0:1.0:-1092.0 - the cut version of La2004 time series
-1240.0:1.0:-1092.0 - is now the same as the common grid
```

Out[77]:



- 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 [13]:

```
# 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 [14]:

```
# 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.

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

In [15]:

```
#= 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 [16]:

```
?cut_timeinterval_from_uivD
```

search: **cut_timeinterval_from_uivD**

Out[16]:

No documentation found.

`cut_timeinterval_from_uivD` is a `Function`.

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

In [17]:

```
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[17]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 149 uncertain values coupled with 149 uncertain indic
es
```

In [18]:

```
# 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
```

Out[18]:

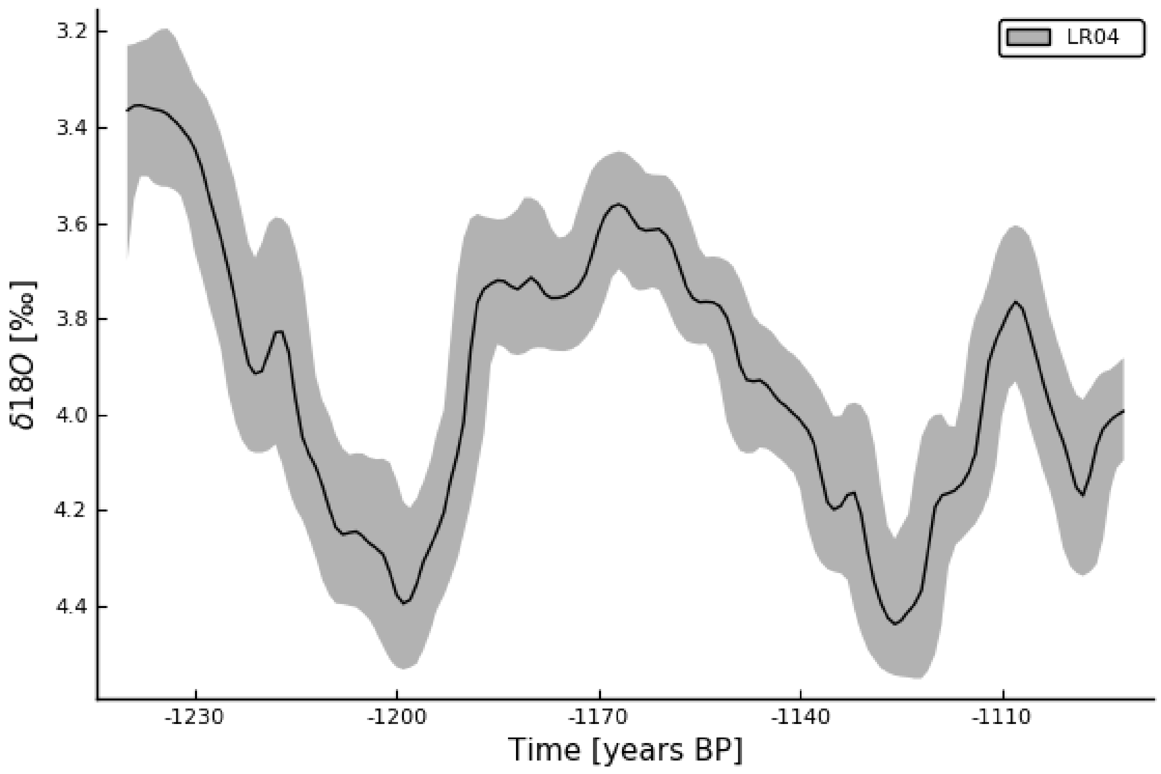
```
-1240.0:1.0:-1092.0
```

In [19]:

```
# 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 [years BP]",
    ylabel =  L"\delta{18}O \ [\perthousand]",
    grid = false, yflip = true
)
```

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

Out[19]:



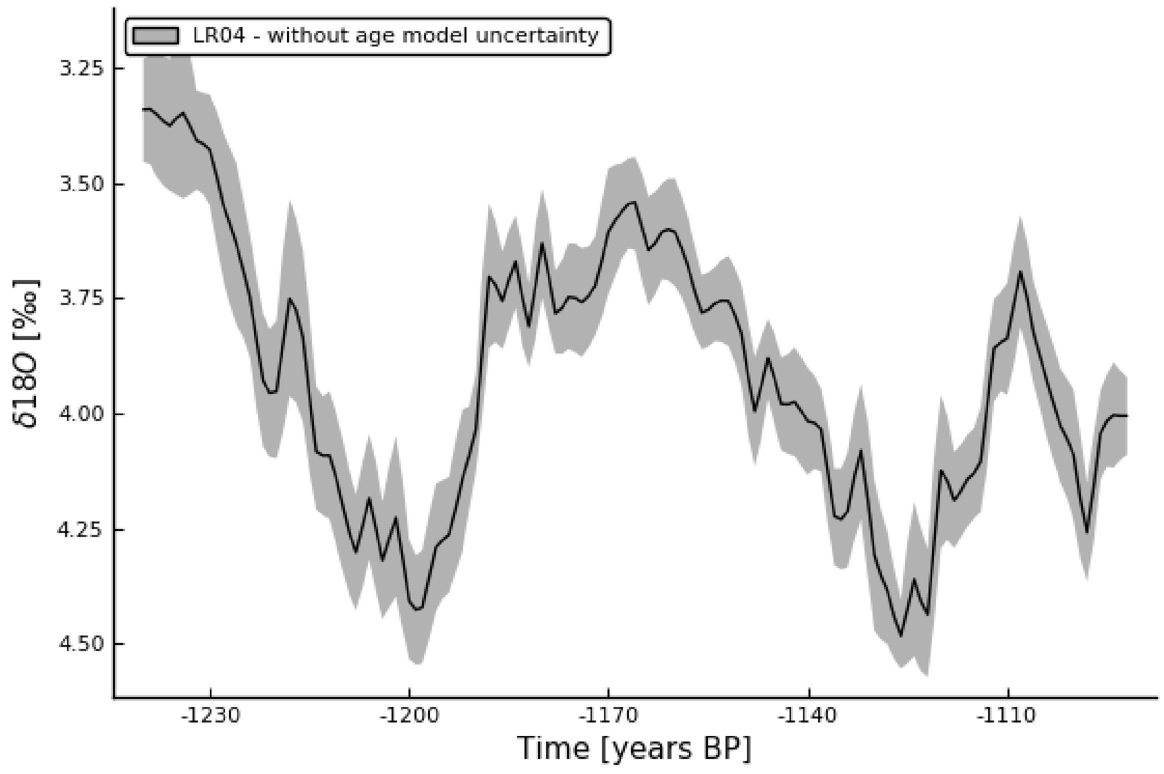
- Out of curiosity, let's see what results we get if we don't include the age model uncertainty

In [20]:

```
LR04 = LR04_binned_fullength_noageunc

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


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


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


# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(LR04_cut_noageunc.values, 0.5)
bin_upperq = quantile.(LR04_cut_noageunc.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(LR04_cut_noageunc.values, 0.025)

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

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

```
Out[20]:
```



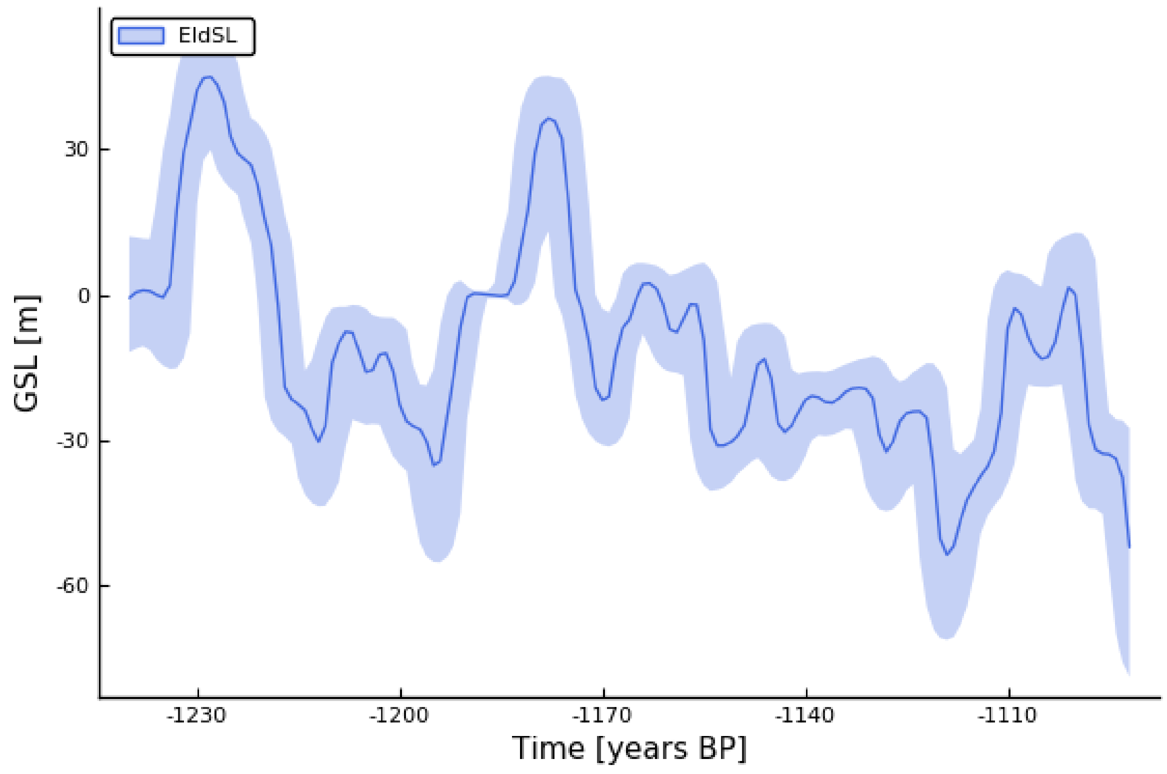
Sea level - Elderfield record

In [76]:

```
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 [years BP]",
    ylabel =  "GSL [m]",
    grid = false
    )
```

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

```
Out[76]:
```



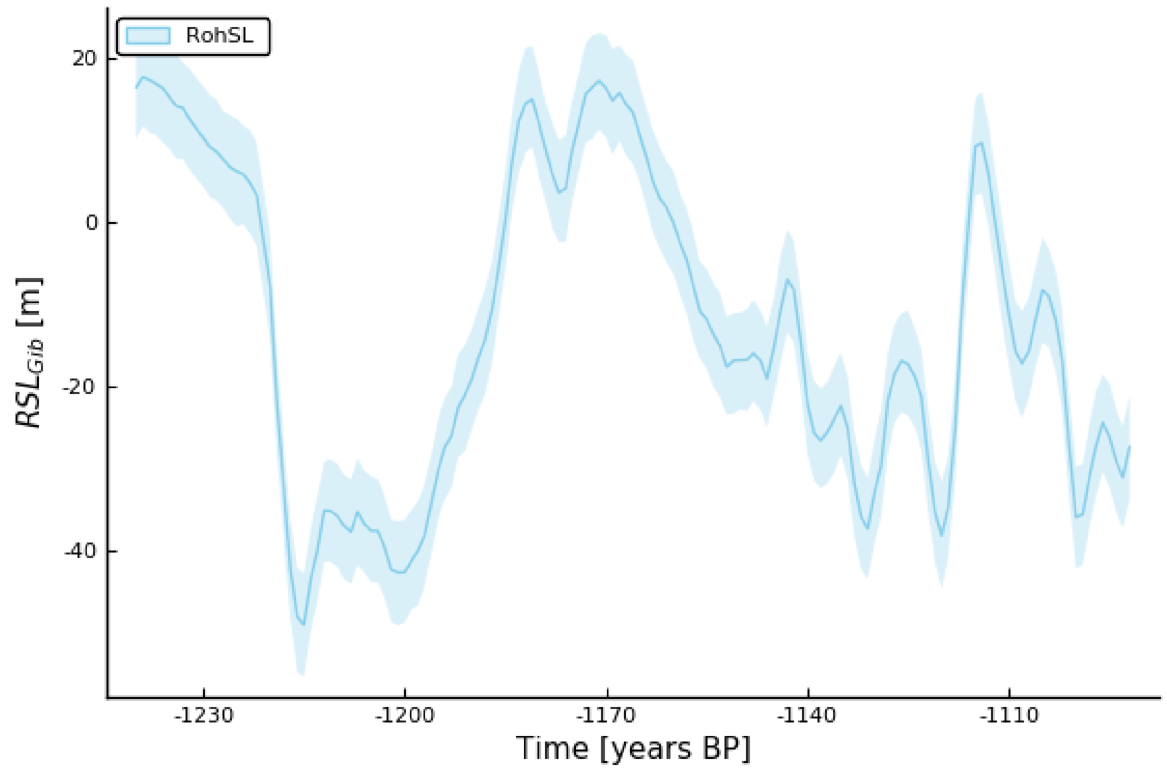
Sea level - Rohling record

In [22]:

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

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

```
Out[22]:
```



pCO2 - Chalk record

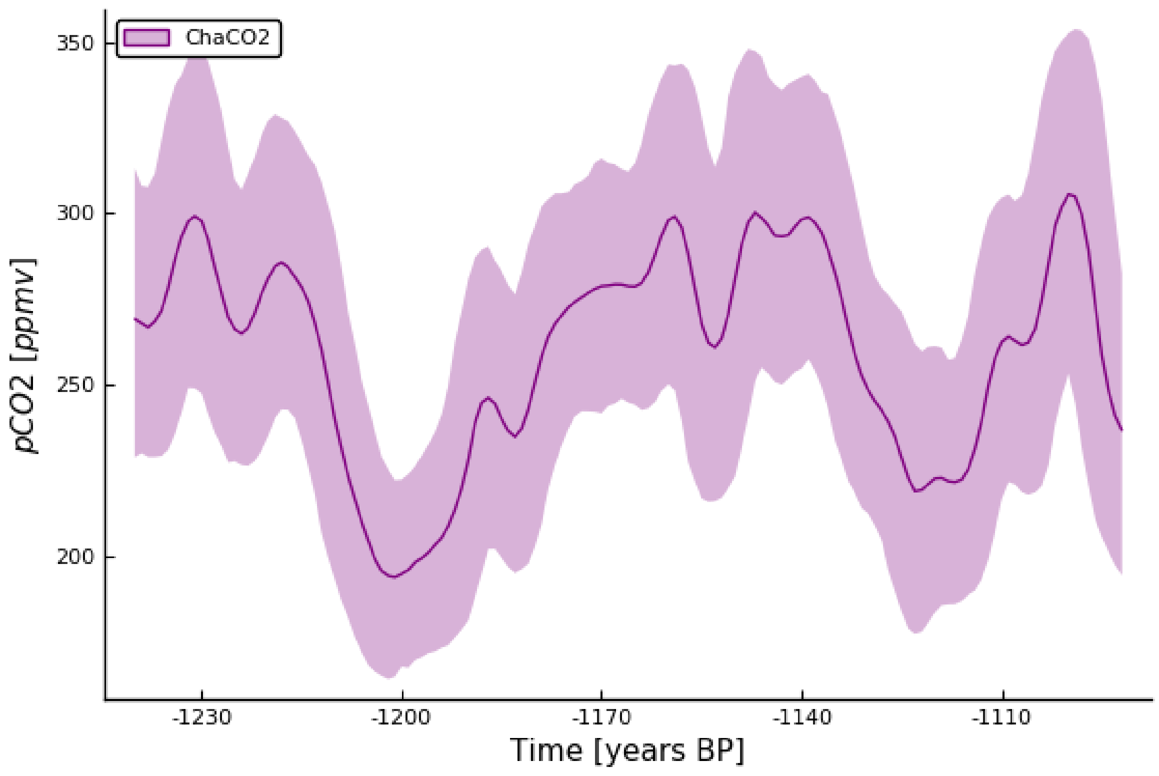
- 1 kyr crid (`C_cut`)

In [23]:

```
ts = C_binned

# 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
C_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_C =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :purple,
    label = "ChaCO2",
    xlabel = "Time [years BP]",
    ylabel = L"pCO2 \ [ppmv]",
    grid = false
    )
```

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

Out[23]:



We also prepare two **high resolution** grids, for the high resolution analysis with La2004 (125 year timestep), and Martinez-García (500 year timestep)

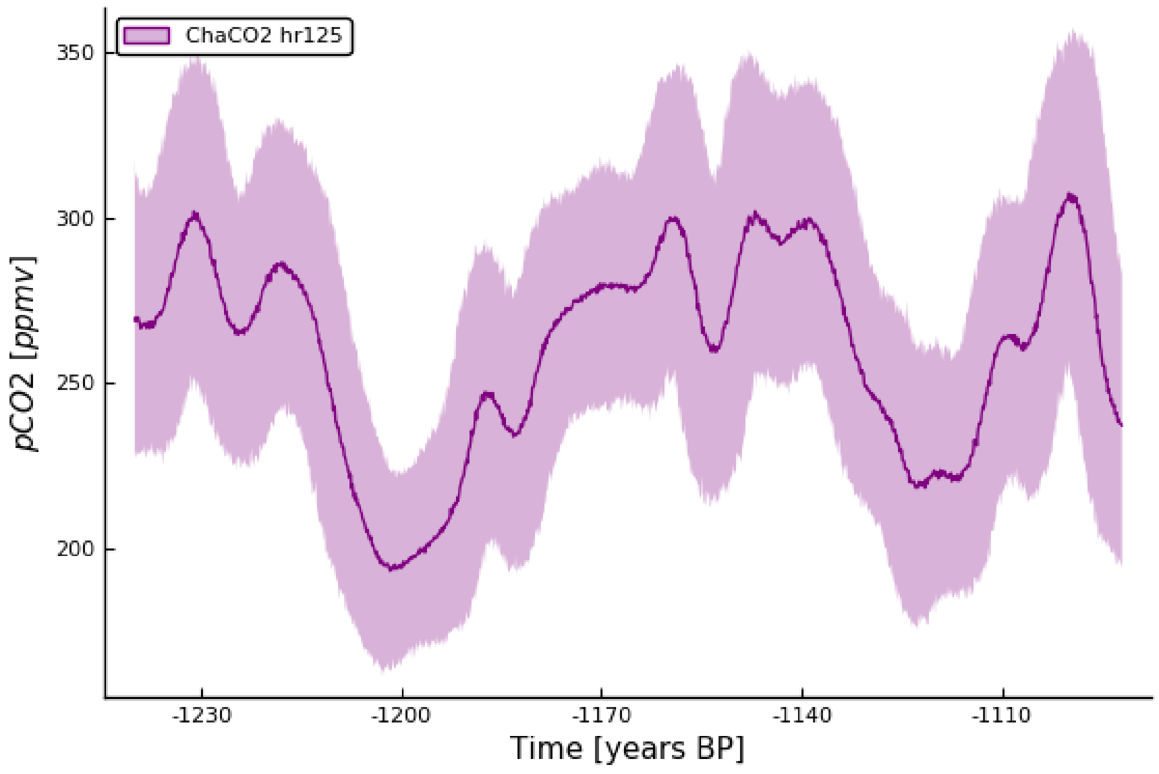
- `C_cut_hr125`

In [24]:

```
# high resolution version
ts = C_binned_hr0125

# 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_hr125
    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
C_cut_hr125 = 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_C_hr125 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :purple,
    label =  "ChaCO2 hr125",
    xlabel =  "Time [years BP]",
    ylabel =  L"pCO2 \ [ppmv]",
    grid = false
    )
```

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

```
Out[24]:
```



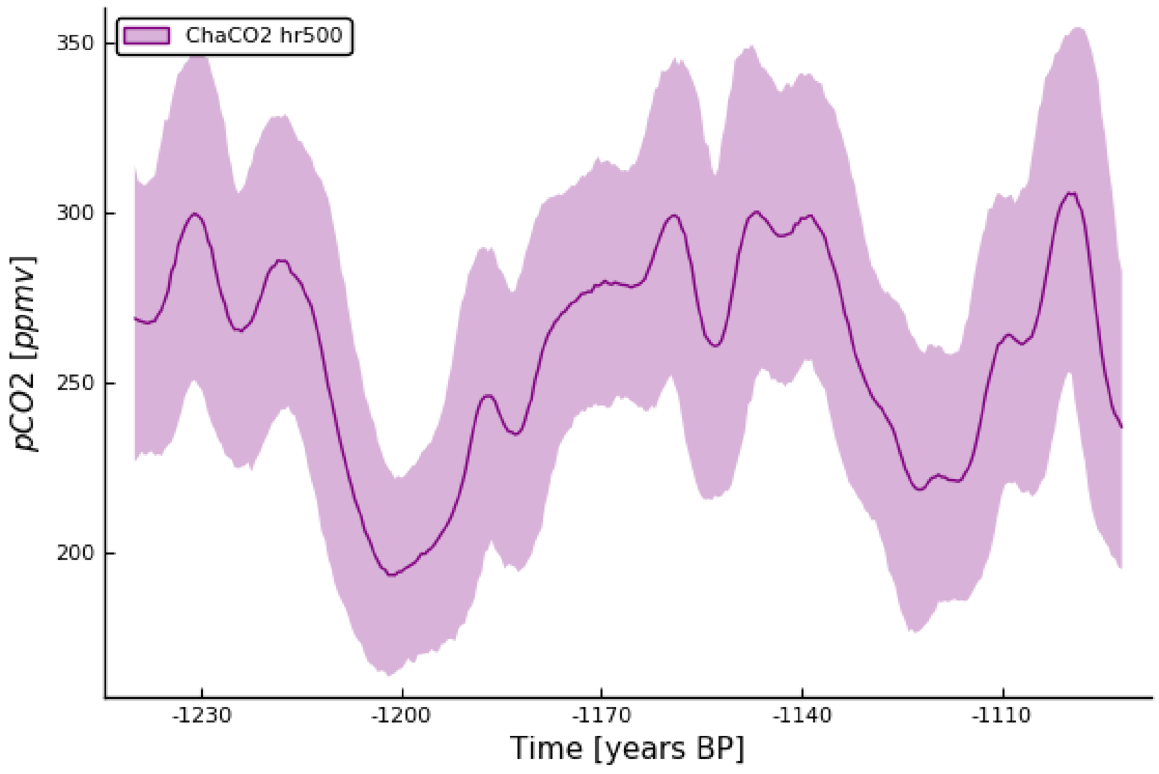
- `C_cut_hr500`

In [25]:

```
# high resolution version
ts = C_binned_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
C_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_C_hr500 =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3, legend = :topleft, color = :purple,
    label =  "ChaCO2 hr500",
    xlabel =  "Time [years BP]",
    ylabel =  L"pCO2 \ [ppmv]",
    grid = false
    )
```

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

```
Out[25]:
```



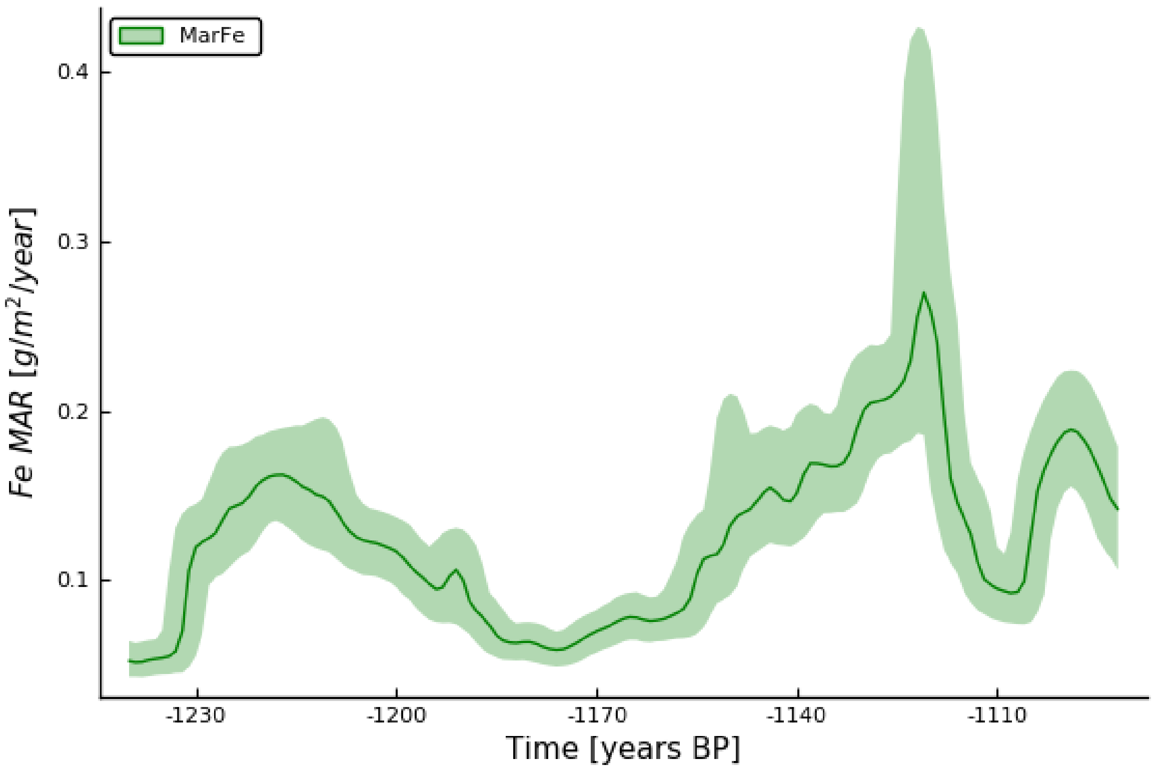
dust - Martinez-García Fe MAR record

In [26]:

```
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 [years BP]",
    ylabel =  L"Fe \ MAR \ [g/m^{2}/year]",
    grid = false
    )
```

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

Out[26]:

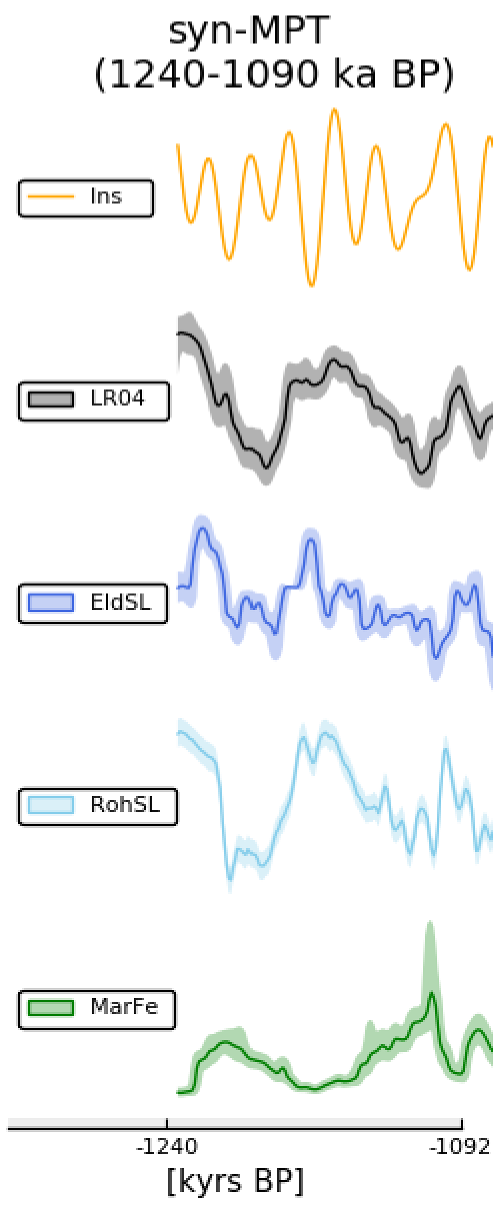


Overview of time series for the syn-MPT interval

In [72]:

```
# overview time series

po0 = plot(plot_La2004, xlabel = "", xaxis = :off, title = "syn-MPT
    (1240-1090 ka BP)")
po1 = plot(plot_LR04, xlabel = "", xaxis = :off)
po3 = plot(plot_E, xlabel = "", xaxis = :off)
po4 = plot(plot_R, xlabel = "", xaxis = :off)
po6 = plot(plot_C, xlabel = "", xaxis = :off)
po8 = plot(plot_MG, xlabel = "", xaxis = :off)
p_xaxis = plot(xlabel = "[kyrs BP]", xlabelfont = 8,  xaxis = :on, xmirror = fal
se)

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

po_alltimeseries = plot(po0,po1,po3,po4,po8, p_xaxis,
    layout = l, #grid(7,1),
    size = (250, 600),
    xlims = (gridstart-80, gridend), xticks = (gridstart : gridend-gridstart  :
gridend),
    ylabel = "", ytickfont = false, yaxis = :off,
    legend = :left, bg_legend = :white)
```

Out[72]:



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**

- Elderfield - La2004
- Rohling - La2004

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

- Elderfield - Chalk
- Rohling - Chalk

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

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

1. **pCO2 - dust**

- Chalk - Martinez-García

1. **pCO2 - insolation**

- Chalk - La2004

1. **dust - insolation**

- Martinez-García - La2004

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

LR04 - La2004

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

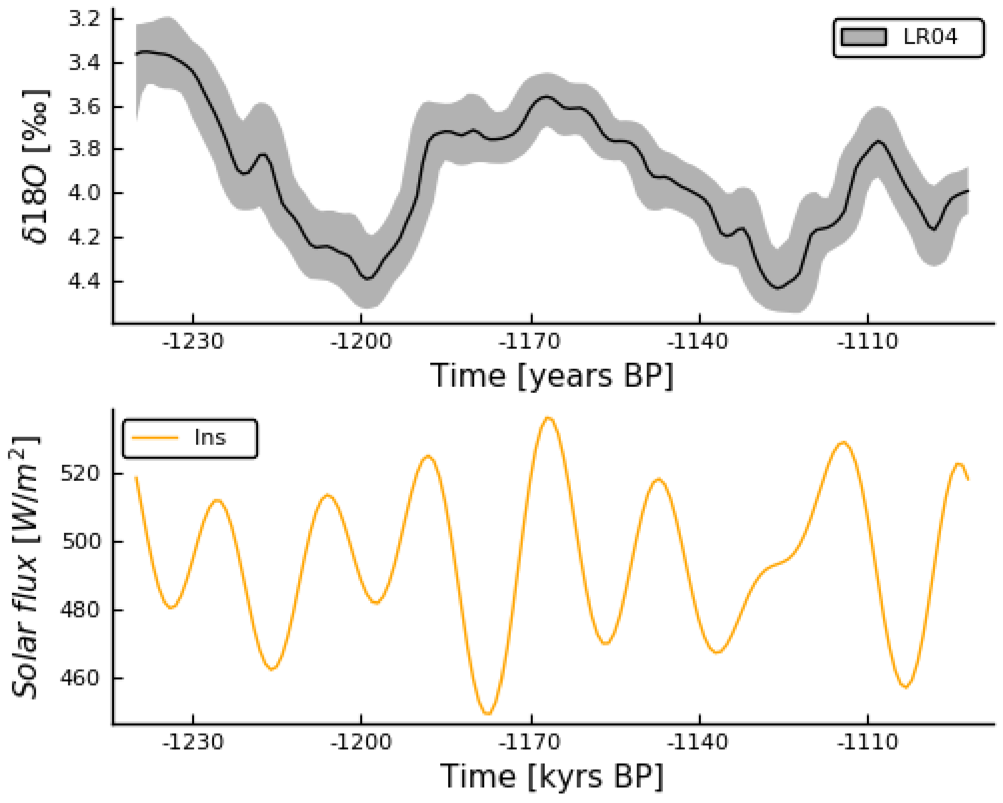
(Both the Elderfield and the Spratt-Lisiecki sea-level records are tuned to the LR04, whose age model is based on a lag of $\delta^{18}O$ behind insolation. This is based on the understanding of the the Milankovitch cycles as a key driver behind the ice age cycles, but is however, a caveat for our method,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

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

In [27]:

```
# 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[27]:



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

In [26]:

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

# First, 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 = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/LR04_La2004.jld2"
    )
```

```
Results are saved in the .jld2 file.  11.122121 seconds (64.55 M all
ocations: 4.786 GiB, 19.13% gc time)
```

In [28]:

```
# Load and normalize the results for comparability

@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/LR04_La2
004.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5);
```

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

In [29]:

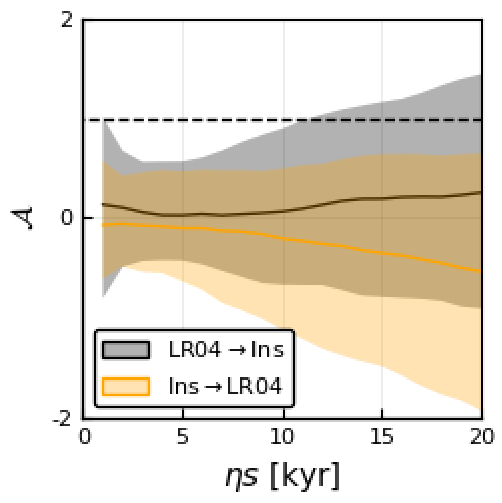
```
### Calculate the quantiles for the confidence interval we want to plot (2σ, aka
95%)

    # ... 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 plot

In [33]:

```
# 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 = (-2,2),
    yticks = (-10:2:10),
    legend = :bottomleft, # WHY DOESN'T THE LEGEND kwarg WORK?
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized # define the predictive asymmetry plo
t...
    size = (250,250), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  "" # mean TE
    )
plot!(normPA_XtoY_median, # ...from X to Y
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("LR04", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :black
    )
plot!(normPA_YtoX_median, # ...from Y to X
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "LR04"),
    fillalpha = 0.3,  color = :orange
    )
```

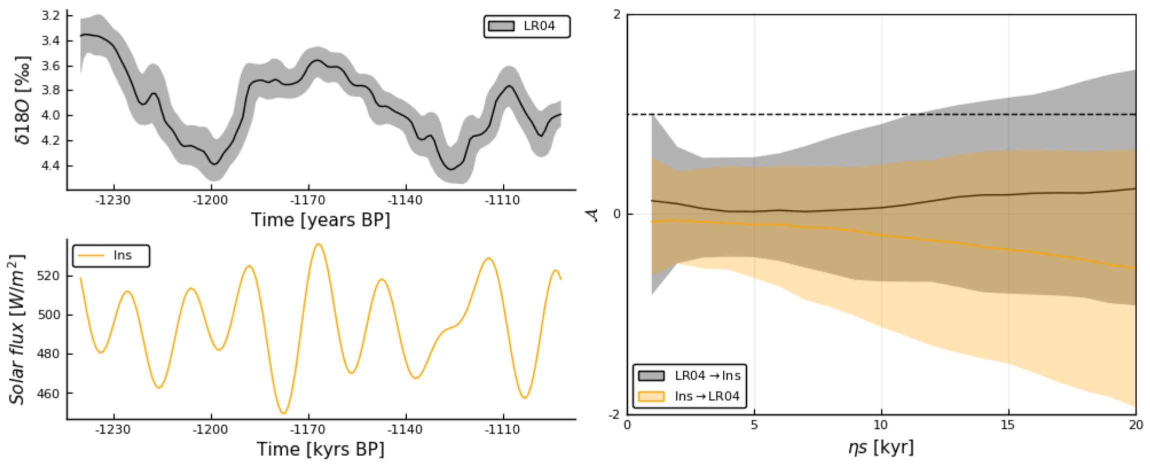
Out[33]:



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

In [34]:

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

plot_results_LR04_La2004 =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_LR04_La2004,
    plot_normPA_LR04_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/LR04_
La2004.pdf") # This is the version with age model uncertainty
```



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

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

Elderfield - La2004

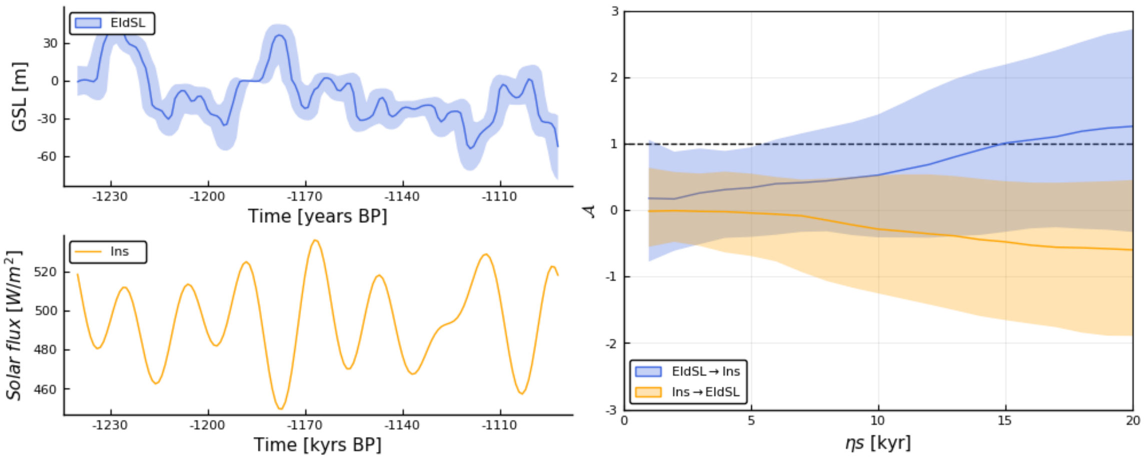
In [74]:

```
# 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 = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/E_La2004.jld2")
```

```
Results are saved in the .jld2 file.  13.235319 seconds (64.25 M all
ocations: 4.765 GiB, 17.72% gc time)
```

In [78]:

```
# Load and normalize the predictive asymmetry results

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/E_La200
4.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5);

# 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 = (-3,3),
    yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label = "" # mean TE
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("EldSL", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :royalblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "EldSL"),
    fillalpha = 0.3,  color = :orange
    )
;


# join plots of time series and pa results to a results subplot
plot_overview_E_La2004 =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_E,
    plot_La2004
    )

plot_results_E_La2004 =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_E_La2004,
    plot_normPA_E_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/E_La2
004.pdf")
```



Rohling - La2004

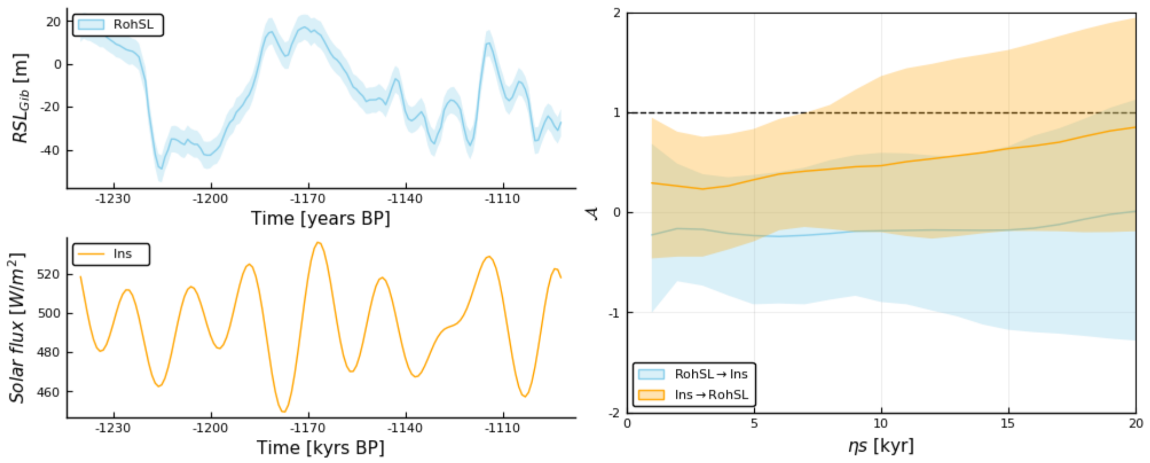
In [33]:

```
# 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 = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/R_La2004.jld2")
```

```
Results are saved in the .jld2 file.  10.754739 seconds (62.94 M all
ocations: 4.677 GiB, 19.75% gc time)
```

In [36]:

```
# Load and normalize the predictive asymmetry results

# Load and normalize the results
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/R_La200
4.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# 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 = (-2,2), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"), # prediction lags
    ylabel =  L"\mathcal{A}", # normalized predictive asymmetry...
    size = (250,250), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  "" # mean TE
    )
plot!(normPA_XtoY_median, # ...from X to Y
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("RohSL", L"\rightarrow", "Ins"),
    fillalpha = 0.3, color = :skyblue
    )
plot!(normPA_YtoX_median, # ...from Y to X
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("Ins", L"\rightarrow", "RohSL"),
    fillalpha = 0.3, color = :orange)
;


# join plots of time series and pa results to a results subplot
plot_overview_R_La2004 =
plot(layout = grid(2,1),
    size = (800, 400),
    plot_R,
    plot_La2004
    )

plot_results_R_La2004 =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_R_La2004,
    plot_normPA_R_La2004)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/R_La2
004.pdf")
```



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

LR04 $\delta^{18}O$ - Chalk pCO2

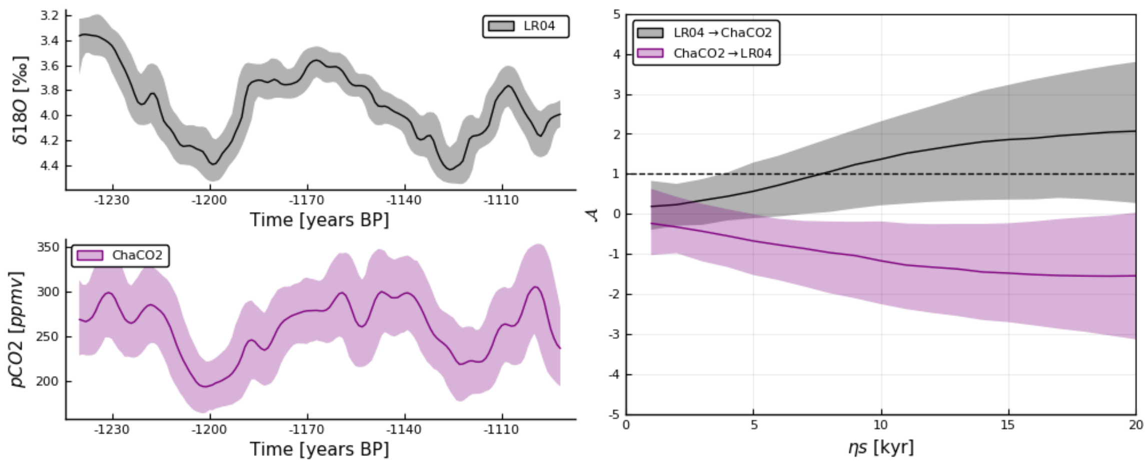
In [35]:

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

# Compute the predictive asymmetry between the two time series (function detaile
d in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 4,
filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/LR0
4_C.jld2" )
```

```
Results are saved in the .jld2 file.  11.251700 seconds (64.58 M all
ocations: 4.736 GiB, 19.04% gc time)
```

In [37]:

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

# load and normalize the predictive asymmetry results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/LR04_C.j
ld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# 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_C =
plot(#title = L"$\mathcal{A}$ between sea level and pCO2",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :topleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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", "ChaCO2"),
    fillalpha = 0.3, legend = :topleft, color = :black
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("ChaCO2", L"\rightarrow", "LR04"),
    fillalpha = 0.3, legend = :topleft, color = :purple)
;


# join plots of time series and pa results to a results subplot
plot_overview_LR04_C =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_LR04,
    plot_C)

plot_results_LR04_C =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_LR04_C,
    plot_normPA_LR04_C)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/LR04_
C.pdf")
```



Elderfield GSL - Chalk pCO2

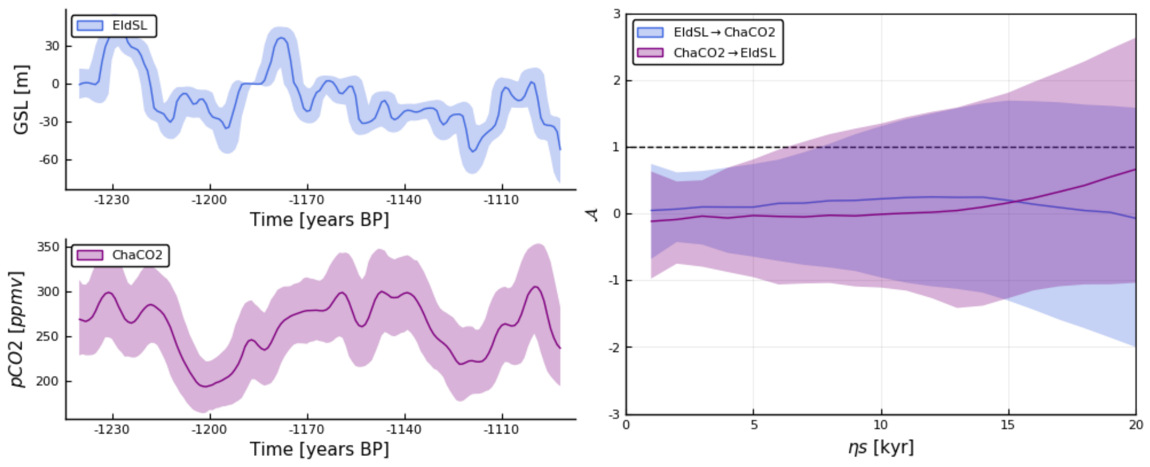
In [37]:

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

# Compute the predictive asymmetry between the two time series (function detaile
d in NB3)
computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/E_C.jld2" )
```

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

In [38]:

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

# load and normalize the predictive asymmetry results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/E_C.jld
2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# 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
;


# 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_C =
plot(#title = L"$\mathcal{A}$ between sea level and pCO2",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-3,3), yticks = (-5:1:5),
    legend = :topleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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", "ChaCO2"),
    fillalpha = 0.3, legend = :topleft, color = :royalblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("ChaCO2", L"\rightarrow", "EldSL"),
    fillalpha = 0.3, legend = :topleft, color = :purple)
;


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

plot_results_E_C =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_E_C,
    plot_normPA_E_C)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/E_C.p
df")
```



Rohling RSL - Chalk pCO2

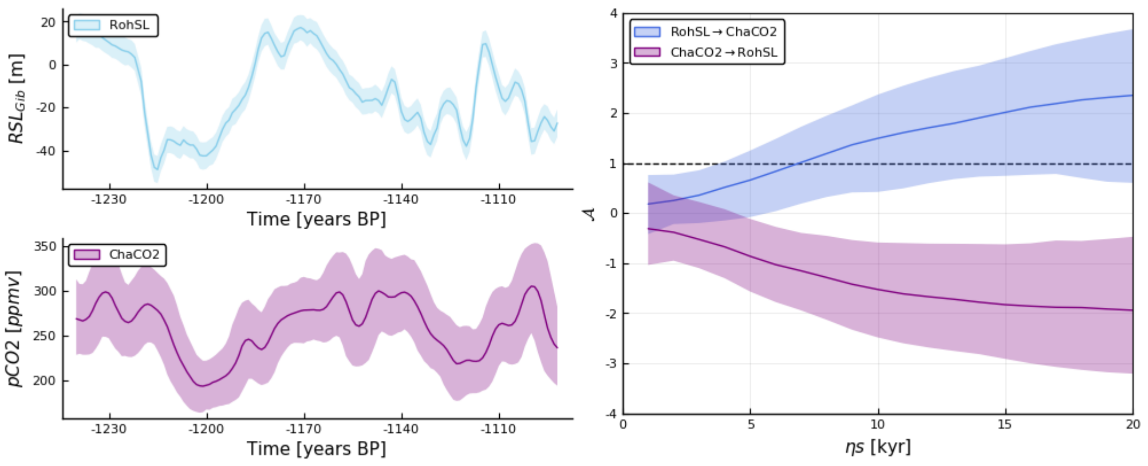
In [39]:

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

# Compute the predictive asymmetry between the two time series (function detaile
d in NB3)
computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/R_C.jld2" )
```

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

In [39]:

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

# read in the results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/R_C.jld
2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# 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_C =
plot(#title = L"$\mathcal{A}$ between sea level and pCO2",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-4,4), yticks = (-5:1:5),
    legend = :topleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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", "ChaCO2"),
    fillalpha = 0.3, legend = :topleft, color = :royalblue
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("ChaCO2", L"\rightarrow", "RohSL"),
    fillalpha = 0.3, legend = :topleft, color = :purple)
;


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

plot_results_R_C =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_R_C,
    plot_normPA_R_C)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/R_C.p
df")
```



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

LR04 - Martinez-García

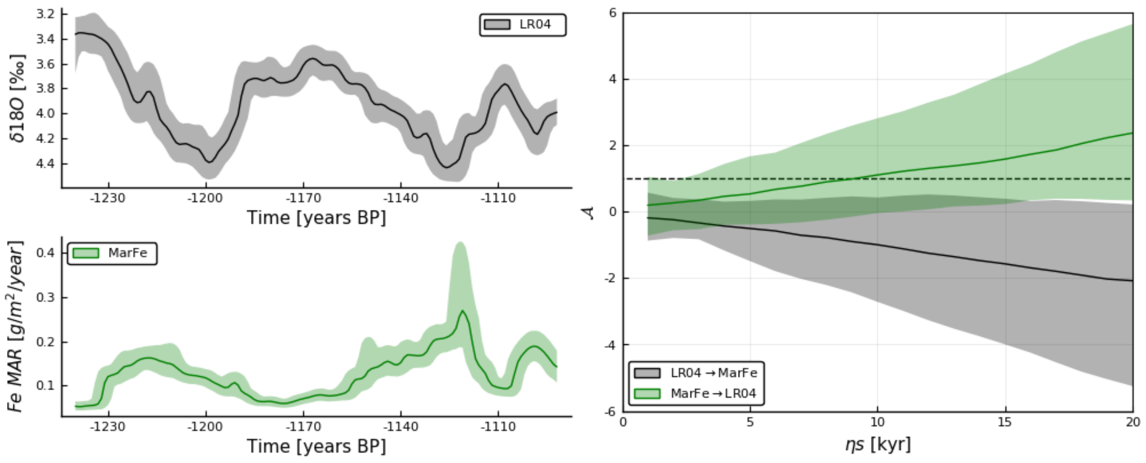
In [41]:

```
#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 = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/LR04_MG.jld2")
```

```
Results are saved in the .jld2 file.  11.995920 seconds (65.06 M all
ocations: 4.771 GiB, 18.10% gc time)
```

In [40]:

```
# Load and normalize the predictive asymmetry results

@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/LR04_MG.
jld2"
# unpack from structure
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# 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 = (-6,6), yticks = (-10:2:10),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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 = (1000, 400),
    plot_LR04,
    plot_MG
    )

plot_results_LR04_MG =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_LR04_MG,
    plot_normPA_LR04_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/LR04_
MG.pdf")
```



Elderfield - Martinez-García

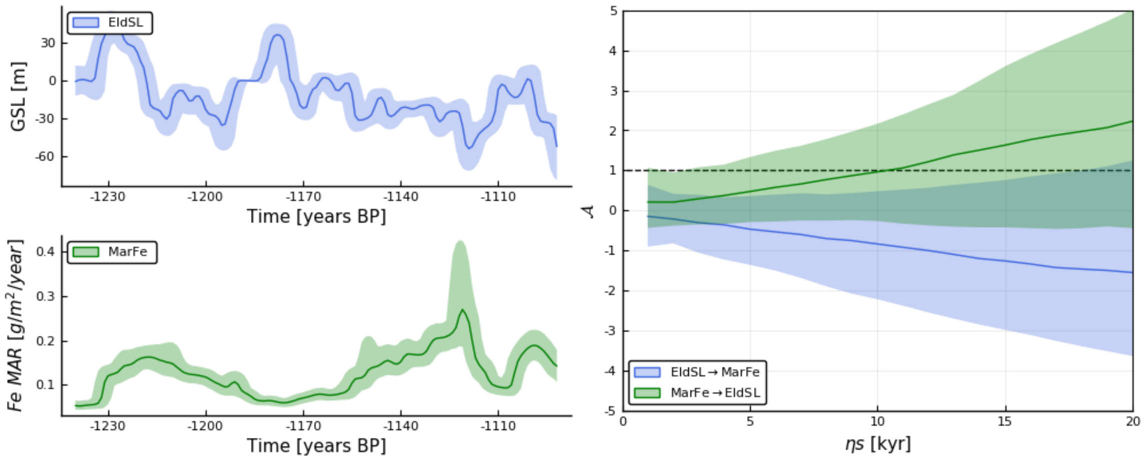
In [43]:

```
# 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 = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/E_MG.jld2")
```

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

In [41]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/E_MG.jld
2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# calculate the quantiles for the 95% confidence interval
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_E_MG =
plot(#title = L"$\mathcal{A}$ between GSL and Southern Ocean Fe deposition",
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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 = (1000, 400),
    plot_E,
    plot_MG
    )

plot_results_E_MG =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_E_MG,
    plot_normPA_E_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/E_MG.
pdf")
```



Rohling - Martinez-García

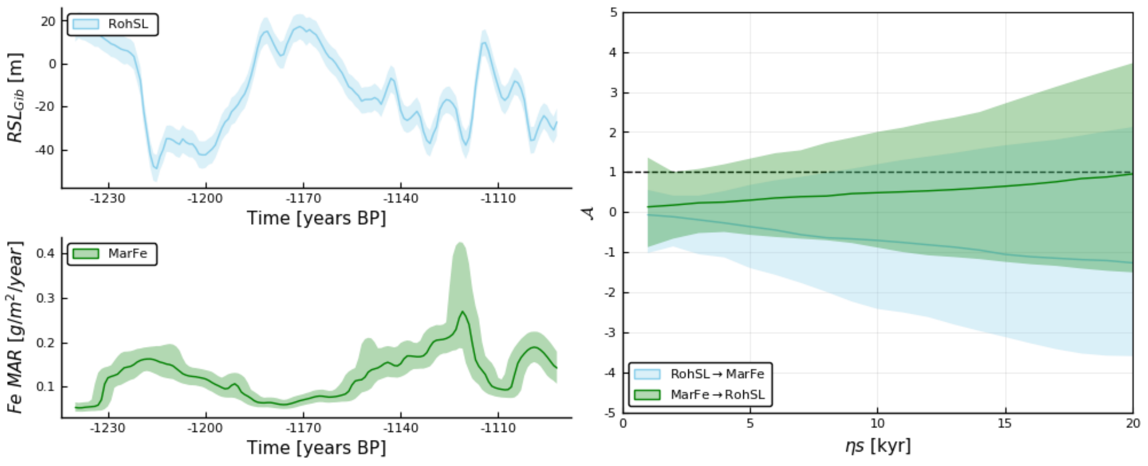
In [45]:

```
#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 = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/R_MG.jld2")
```

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

In [42]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/R_MG.jld
2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# calculate the quantiles for the 95% confidence interval
    # ... for PA from X to Y
normPA_XtoY_median = [quantile(normPA_XtoY[i,:], 0.5) for i in 1:ηmax]
# median
normPA_XtoY_upper = [quantile(normPA_XtoY[i,:], 0.975) for i in 1:ηmax] .- normP
A_XtoY_median # upper quantile
normPA_XtoY_lower = normPA_XtoY_median .- [quantile(normPA_XtoY[i,:], 0.025) for
i in 1:ηmax] # lower quantile
    # ... for PA from Y to X
normPA_YtoX_median = [quantile(normPA_YtoX[i,:], 0.5) for i in 1:ηmax]
# median
normPA_YtoX_upper = [quantile(normPA_YtoX[i,:], 0.975) for i in 1:ηmax] .- normP
A_YtoX_median # upper quantile
normPA_YtoX_lower = normPA_YtoX_median .- [quantile(normPA_YtoX[i,:], 0.025) for
i in 1:ηmax] # lower quantile
;


# defining the results plot
plot_normPA_R_MG =
plot(xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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 = (1000, 400),
    plot_R,
    plot_MG
    )

plot_results_R_MG =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_R_MG,
    plot_normPA_R_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/R_MG.
pdf")
```



5. pCO2 - dust

Chalk - Martinez-García

In [47]:

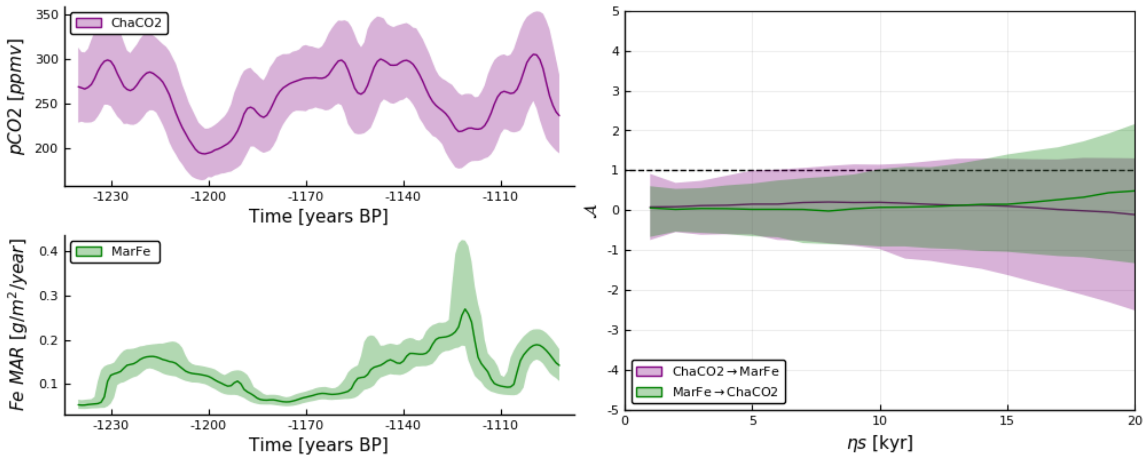
```
# recall the time series on the common grid as X and Y
X = C_cut
Y = MG_cut

# Compute the predictive asymmetry (function defined in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/C_MG.jld2" )
```

```
Results are saved in the .jld2 file.  13.670611 seconds (64.37 M all
ocations: 4.721 GiB, 17.04% gc time)
```

In [43]:

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

@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/C_MG.jld
2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# 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_C_MG =
plot(#title = L"$\mathcal{A}$ between #sea level and #insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), border = true,
    grid = true,
    hline([1], line = (:dash, :black)),
    label =  ""
    )
plot!(normPA_XtoY_median,
    ribbon = (normPA_XtoY_lower, normPA_XtoY_upper),
    label =  string("ChaCO2", L"\rightarrow", "MarFe"),
    fillalpha = 0.3,  color = :purple
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("MarFe", L"\rightarrow", "ChaCO2"),
    fillalpha = 0.3, color = :green)


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

```
plot_results_C_MG =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_C_MG,
    plot_normPA_C_MG)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/C_MG.
pdf")
```



- High resolution analysis?

MarFe mean resoulution for thie synMPT_Cgrid_e4_La2004nB time interval was one observation every 680 years. We have therefore decided not to run analyses with a higher resolution time step for this time interval.

---

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

La2004 - Chalk

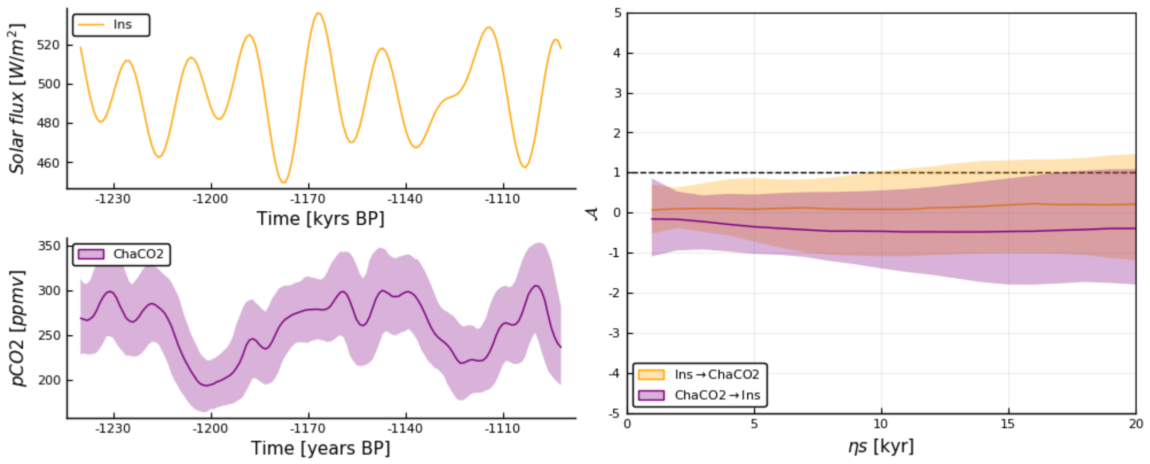
In [49]:

```
### 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 = C_cut

# Compute the predictive asymmetry (function detailed in NB3)
computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 4,
    filepath =  "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/La2004_C.jld2")
```

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

In [44]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/La2004_
C.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)

# 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_C =
plot(#title = L"$\mathcal{A}$ between pCO2 and insolation"
    xlims = (0, ηmax),
    xticks = (0 : 5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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", "ChaCO2"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("ChaCO2", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :purple)
;


# join plots of time series and pa results to a results subplot
plot_overview_La2004_C =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_La2004,
    plot_C)

plot_results_La2004_C =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_La2004_C,
    plot_normPA_La2004_C)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/La200
4_C.pdf")
```



Sensitivity analysis on higher resolution time steps

- High resolution analysis with 125 year time step

Note: longer time series, so we lower the false positive rate *f* = 1

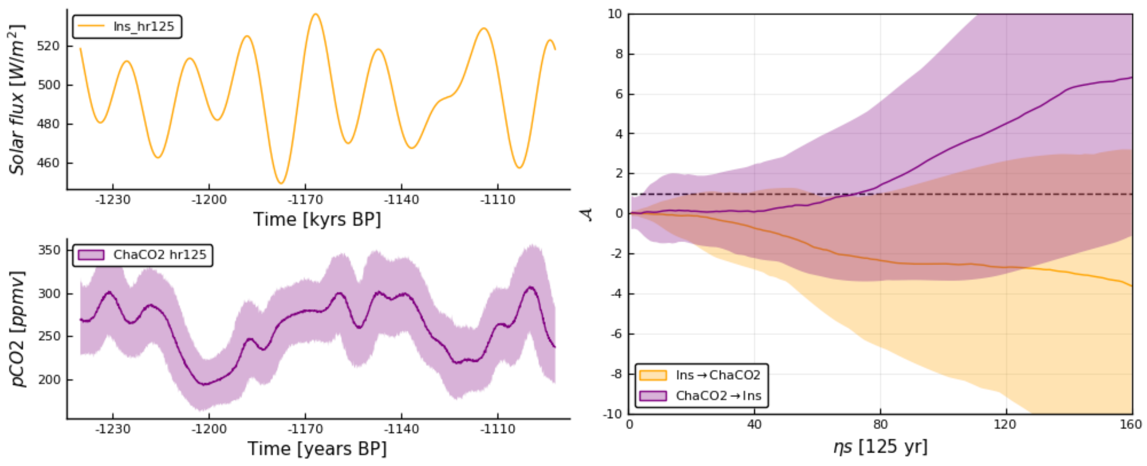
In [51]:

```
# recall the time series on the common grid as X and Y
X = La2004_insol_cut_hr125
Y = C_cut_hr125

# Compute the predictive asymmetry (function detailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 0.125, ηmax = Int(20/0.125),
#\epsilon defined by the palus horizon (= 6)
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/La2004_C_hr125.jld2")
```

```
Results are saved in the .jld2 file. 620.061617 seconds (3.85 G allo
cations: 275.385 GiB, 19.61% gc time)
```

In [45]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/La2004_C
_hr125.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_C_hr125 =
plot(#title = L"$\mathcal{A}$ between pCO2 and insolation"
    xlims = (0, ηmax),
    xticks = (0 : 8*5 : ηmax),
    ylims = (-10,10), yticks = (-20:2:20),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [125 yr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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", "ChaCO2"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("ChaCO2", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :purple)
;


# join plots of time series and pa results to a results subplot
plot_overview_La2004_C_hr125 =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_La2004_hr125,
    plot_C_hr125)

plot_results_La2004_C_hr125 =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_La2004_C_hr125,
    plot_normPA_La2004_C_hr125)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/La200
4_C_hr125.pdf")
```



> A false positive! This may well have connection ot an oversampling of the dataset (Chalk et al report the data with a resolution of 125 yrs, while original sampling was every 3-4-4.5 kyrs, i.e. an order of magnitude higher...)

Let's do a little sensitivity analysis with the time step.

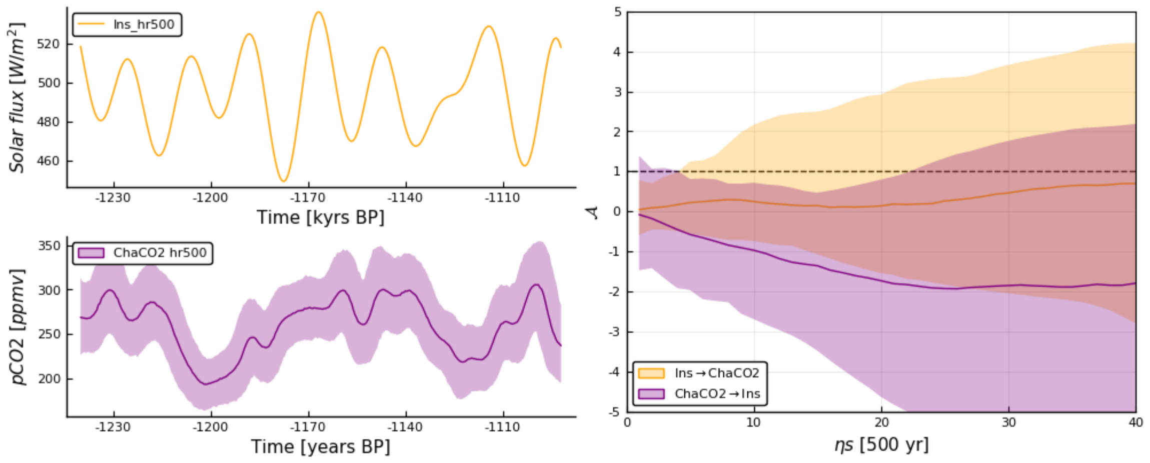
In [53]:

```
# recall the time series on the common grid as X and Y
X = La2004_insol_cut_hr500
Y = C_cut_hr500

# Compute the predictive asymmetry (function detailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 0.5, ηmax = Int(20/0.5),
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/La2004_C_hr500.jld2")
```

```
Results are saved in the .jld2 file.  39.382083 seconds (242.40 M al
locations: 17.835 GiB, 19.48% gc time)
```

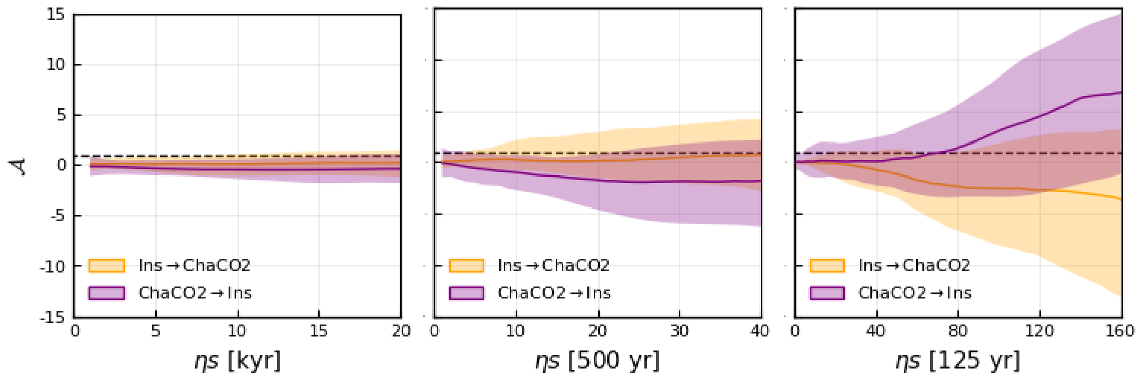
In [46]:

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

# Load and normalize the predictive asymmetry results
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/La2004_C
_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_C_hr500 =
plot(#title = L"$\mathcal{A}$ between pCO2 and insolation"
    xlims = (0, ηmax),
    xticks = (0 : 2*5 : ηmax),
    ylims = (-5,5), yticks = (-5:1:5),
    legend = :bottomleft,
    xlabel =  string(L"ηs"," [500 yr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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", "ChaCO2"),
    fillalpha = 0.3, color = :orange
    )
plot!(normPA_YtoX_median,
    ribbon = (normPA_YtoX_lower, normPA_YtoX_upper),
    label =  string("ChaCO2", L"\rightarrow", "Ins"),
    fillalpha = 0.3,  color = :purple)
;


# join plots of time series and pa results to a results subplot
plot_overview_La2004_C_hr500 =
plot(layout = grid(2,1),
    size = (1000, 400),
    plot_La2004_hr500,
    plot_C_hr500)

plot_results_La2004_C_hr500 =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_La2004_C_hr500,
    plot_normPA_La2004_C_hr500)

savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/La200
4_C_hr500.pdf")
```



In [49]:

```
pAA_C_hr500 = plot(plot_normPA_La2004_C_hr500, ylabel = "", ytickfont = false)
pAA_C_hr125 = plot(plot_normPA_La2004_C_hr125, ylabel = "", ytickfont = false)

plot(plot_normPA_La2004_C, pAA_C_hr500, pAA_C_hr125,
    layout = grid(1,3), size = (750,250),
    ylims = (-15,15), yticks = (-20:5:20),
    bg_legend = :transparent
    )

savefig("../../results_ePalus_ns20/sensitivity_analyses/hr_insol-co2_synMPT_Cgri
d_e4_La2004nB.pdf")
```



False positive for analysis with shorter time step. **Discussion.** The false positive might stem from over-sampling of the data: The dataset was reported as a probabilistic analysis with a resolution of 125 years between each observation - however, Chalk et al. (2017) report an original sampling of one value every 3.5-4.5 kyrs, which is an order of magnitude higher.

This goes to show must be mindful when interpolating data for use with this method. As a rule of thumb, we have in this project not interpolated to higer resolution than the mean resolution of the original data.

---

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

La2004 - Martinez-García

In [56]:

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


# Compute 150 families of transfer entropy and predictive asymmetry (function de
tailed in NB3)
@time computePredictiveAsymmetries(X, Y, timestep = 1, ηmax = 20, ϵ = 4,
    filepath = "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004n
B/La2004_MG.jld2")
```

```
Results are saved in the .jld2 file.  10.829337 seconds (63.66 M all
ocations: 4.729 GiB, 18.79% gc time)
```

In [50]:

```
f # Back on the 1 kyr grid with shorter time series length - check that we have
 the higher false positive rate again (1.5)
```

Out[50]:

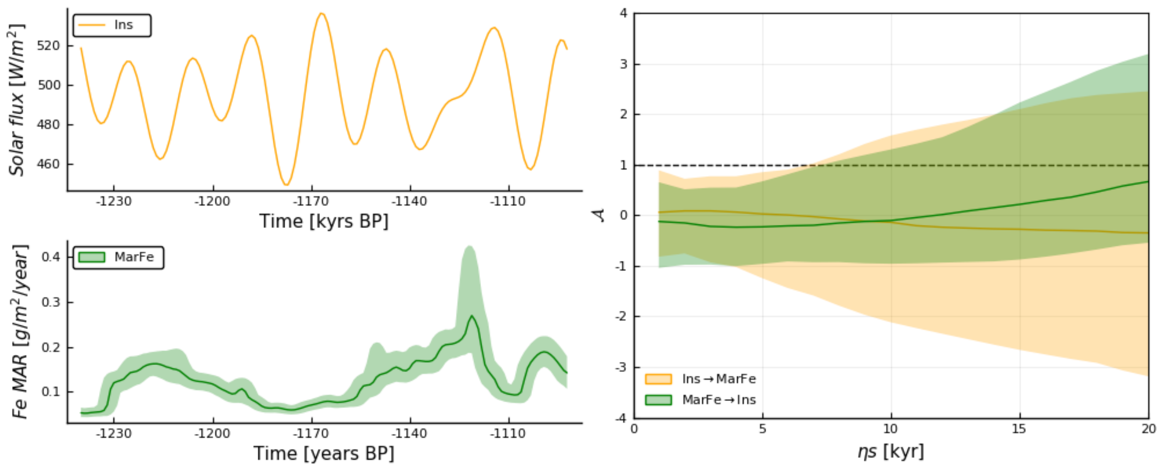
```
1.5
```

In [51]:

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

# load and normalize the predictive asymmetry results computed above
@load "../../results_ePalus_ns20/pa_jld2_files/synMPT_Cgrid_e4_La2004nB/La2004_M
G.jld2"
normPA_XtoY = normalizePredictiveAsymmetry(rsTE_XtoY, rsPA_XtoY, ηmax = ηmax, f
= 1.5)
normPA_YtoX = normalizePredictiveAsymmetry(rsTE_YtoX, rsPA_YtoX, ηmax = ηmax, f
= 1.5)


# 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 = (-4,4), yticks = (-5:1:5),
    legend = :bottomleft, bg_legend = :transparent,
    xlabel =  string(L"ηs"," [kyr]"),
    ylabel =  L"\mathcal{A}",
    size = (250,250), 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 = (1000, 400),
    plot_La2004,
    plot_MG)
```

```
plot_results_La2004_MG =
plot(size = (1000,400),
    layout = grid(1,2),
    plot_overview_La2004_MG,
    plot_normPA_La2004_MG)

#savefig("../../results_ePalus_ns20/pa_ResultPlots/synMPT_Cgrid_e4_La2004nB/La20
04_MG.pdf")
```

Out[51]:



- high resolution version (time step of 500 years)

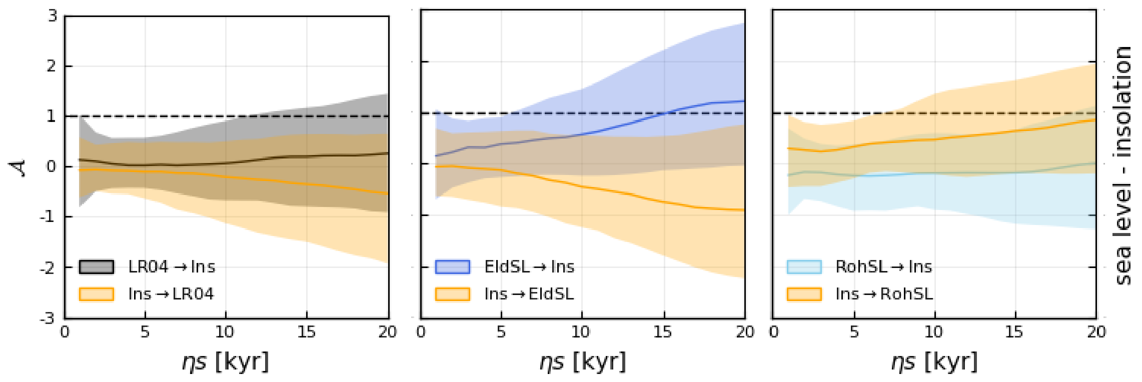
> We decided not to interpolate the Maritnez-Garcia record for this, as the 500 yr time step is higher resolution than the mean resolution across the time interval (680 years)

Ensemble plots of results

1. **GSL - insolation**

In [52]:

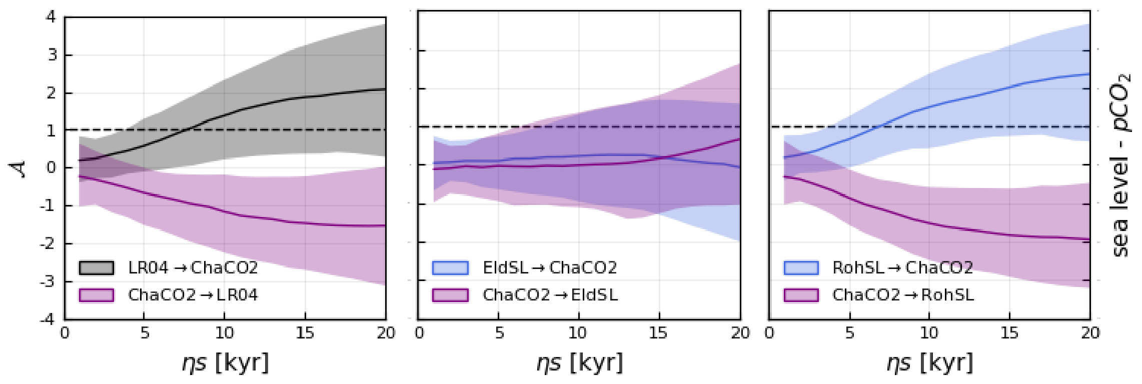
```
# remove the legends from the individual plots

p12 = plot(plot_normPA_LR04_La2004, legend = :bottomleft, bg_legend = :transpare
nt, ylabel = L"\mathcal{A}")
p13 = plot(plot_normPA_E_La2004,    legend = :bottomleft, bg_legend = :transpare
nt, ylabel = "", ytickfont = false, )
p17 = plot(plot_normPA_R_La2004, legend = :bottomleft, bg_legend = :transparent,
ylabel = string("sea level - insolation"), ymirror = true, ytickfont = false)


# plot the 6 results plots and the legend plot to the right
pe_gsl_insol = plot(
    p12, p13, p17,
    layout = grid(1,3), size = (750,250),
    border = true, xaxis = :on,
    ylims = (-3,3), yticks = (-10:1:10),
    )

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/synMPT_Cgrid_e4_La2004n
B/gsl_insol.pdf")
```



1. **GSL - pCO2**

In [53]:

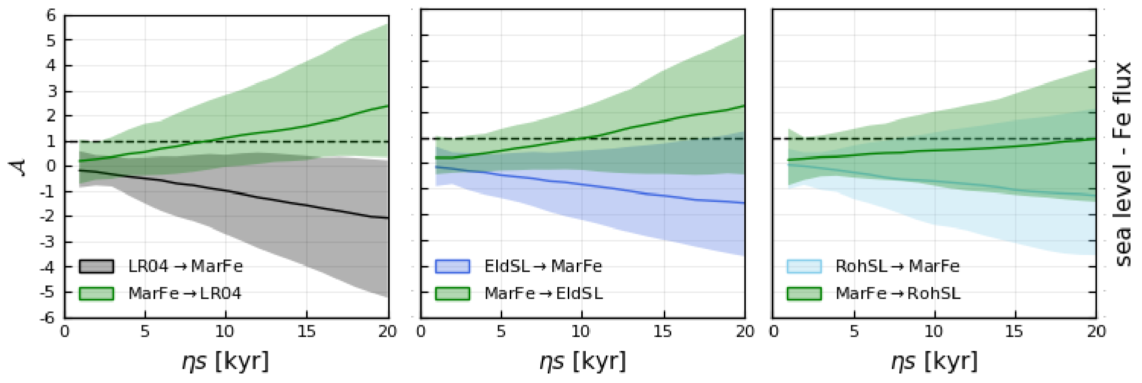
```
p32 = plot(plot_normPA_LR04_C, ytickfont = 8, ylabel = L"\mathcal{A}")
p33 = plot(plot_normPA_E_C, ylabel = "", ytickfont = false)
p37 = plot(plot_normPA_R_C, ylabel = string("sea level - ", L"pCO_{2}"), ymirror
= true, ytickfont = false)


pe_gsl_co2 = plot(
    p32, p33, p37,

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

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/synMPT_Cgrid_e4_La2004n
B/gsl-co2.pdf")
```



1. **GSL - dust**

In [54]:

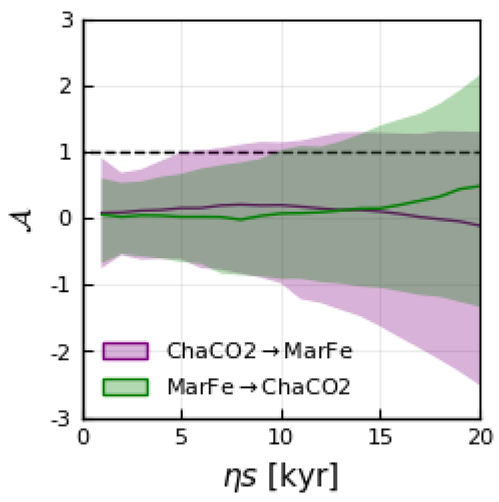
```
p42 = plot(plot_normPA_LR04_MG, ytickfont = 8, ylabel = L"\mathcal{A}")
p43 = plot(plot_normPA_E_MG, ylabel = "", ytickfont = false)
p47 = plot(plot_normPA_R_MG, ylabel = string("sea level - Fe flux"), ymirror = t
rue, ytickfont = false)


pe_gsl_dust = plot(
    p42, p43, p47,
    layout = grid(1,3), size = (750,250),
    border = true, xaxis = :on,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-6,6), yticks = (-10:1:10),
    )

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/synMPT_Cgrid_e4_La2004n
B/gsl-dust.pdf")
```



1. **pCO2 - dust**

In [55]:

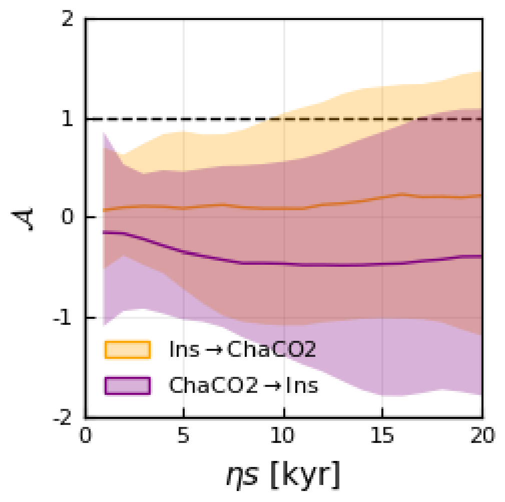
```
pe_co2_dust = plot(
    plot_normPA_C_MG,
    size = (250,250),
    #plot_normPA_C_MG, p_empty, p_empty,
    #layout = (1,3),size = (750,250),
    border = true,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-3,3), yticks = (-10:1:10),
)

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/synMPT_Cgrid_e4_La2004n
B/co2-dust.pdf")
```



1. **pCO2 - insolation**

In [56]:

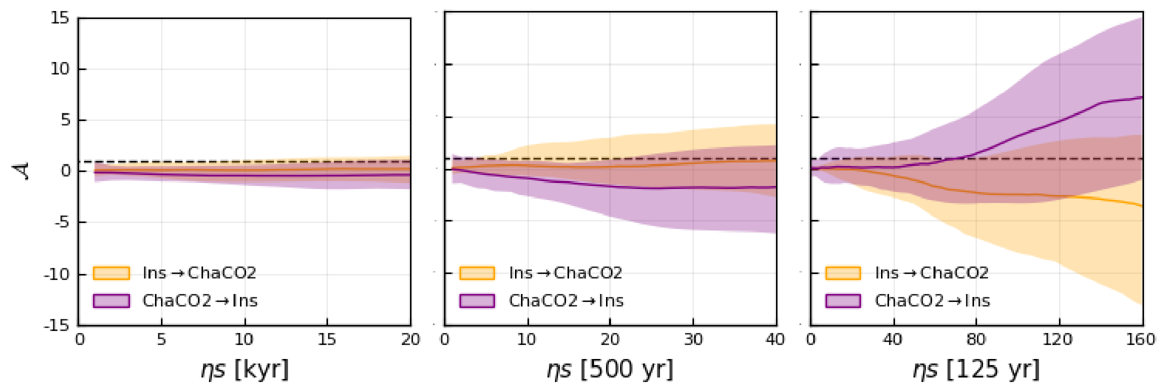
```
p61 = plot(plot_normPA_La2004_C)
p62 = plot(plot_normPA_La2004_C_hr500, ylabel = "", ytickfont = false, yaxis = :
on)
p63 = plot(plot_normPA_La2004_C_hr125, ylabel = "", ytickfont = false, yaxis = :
on)


pe_insol_co2 = plot(
    p61, #p62, p63,
    #layout = grid(1,3),
    size = (250,250),
    border = true, xaxis = :on,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-2,2), yticks = (-20:1:20),

    )

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

plot(pe_insol_co2)

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/synMPT_Cgrid_e4_La2004n
B/insol-co2.pdf")
```



In [57]:

```
p61 = plot(plot_normPA_La2004_C)
p62 = plot(plot_normPA_La2004_C_hr500, ylabel = "", ytickfont = false, yaxis = :
on)
p63 = plot(plot_normPA_La2004_C_hr125, ylabel = "", ytickfont = false, yaxis = :
on)


pe_insol_co2_sensitivity_hr = plot(
    p61, p62, p63,
    layout = grid(1,3),
    size = (750,250),
    border = true, xaxis = :on,
    legend = :bottomleft, bg_legend = :transparent,
    ylims = (-15,15), yticks = (-20:5:20),

    )

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

plot(pe_insol_co2_sensitivity_hr)

#savefig("../../results_ePalus_ns20/sensitivity_analyses/hr_insol-co2_synMPT_Cgr
id_e4_La2004nB.pdf")
```

Out[57]:



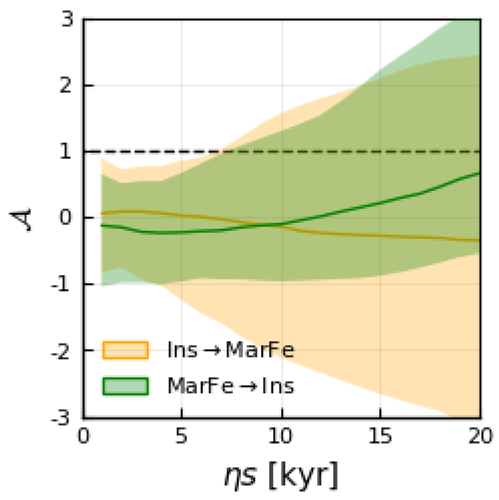
False positive for analysis with shorter time step. **Discussion.** The false positive might stem from over-sampling of the data: The dataset was reported as a probabilistic analysis with a resolution of 125 years between each observation - however, Chalk et al. (2017) report an original sampling of one value every 3.5-4.5 kyrs, which is an order of magnitude higher.

This goes to show must be mindful when interpolating data for use with this method. As a rule of thumb, we have in this project not interpolated to higer resolution than the mean resolution of the original data.

1. **insolation - dust**

In [58]:

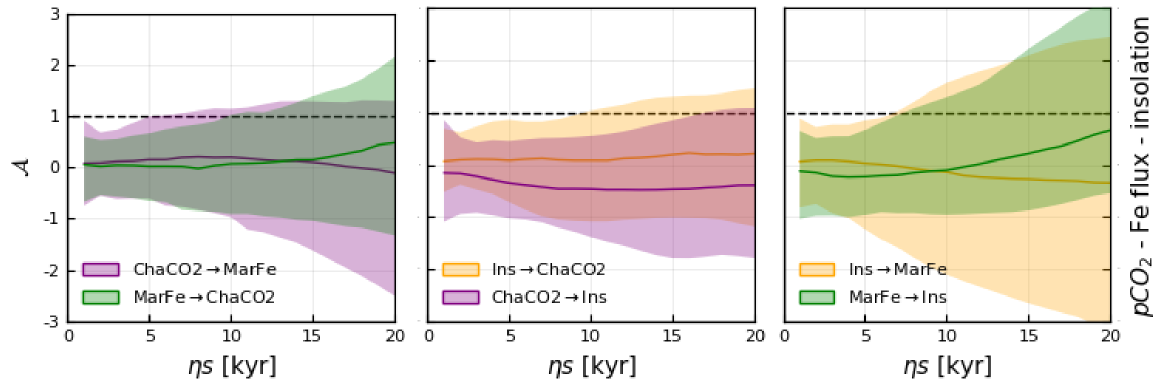
```
p74 = plot(plot_normPA_La2004_MG)
#p75 = plot(plot_normPA_La2004_MG_hr500, ylabel = "", ytickfont = false)

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

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

plot(pe_insol_dust)

savefig("../../results_ePalus_ns20/ensemble_normPA_plots/synMPT_Cgrid_e4_La2004n
B/e_insol_dust_synMPT_Cgrid_e4_La2004nB.pdf")
```



In [59]:

```
p81 = plot(plot_normPA_C_MG, ylabel = L"\mathcal{A}")
p82 = plot(plot_normPA_La2004_C, ylabel = "", ytickfont = false )
p83 = plot(plot_normPA_La2004_MG, ylabel = string(L"pCO_{2}", " - Fe flux - inso
lation"), ymirror = true, ytickfont = false)


pe_fit3 = plot(
    p81, p82, p83,
    layout = grid(1,3), size = (750,250),
    border = :true, xaxis = :on,
    ylims = (-3,3), yticks = (-10:1:10),
    )
```

Out[59]:



Overview: predictive asymmetry for the `synMPT_Cgrid_e4_La2004nB` grid

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

In [61]:

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

Out[61]:

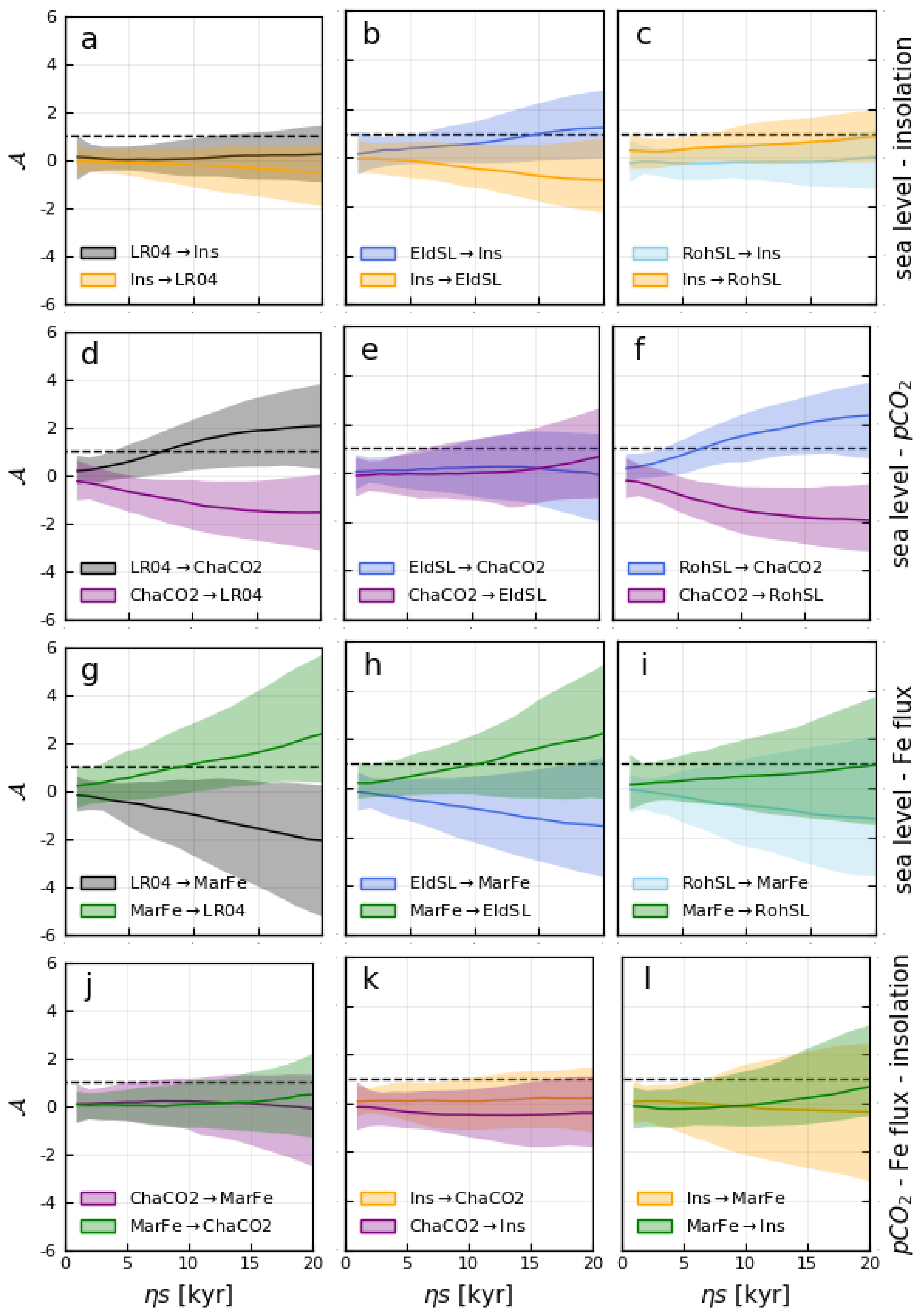
```
26
```

In [62]:

```
p_empty = plot(size = (250,250), border = false, xticks = :none, yticks = :none
);
```

In [63]:

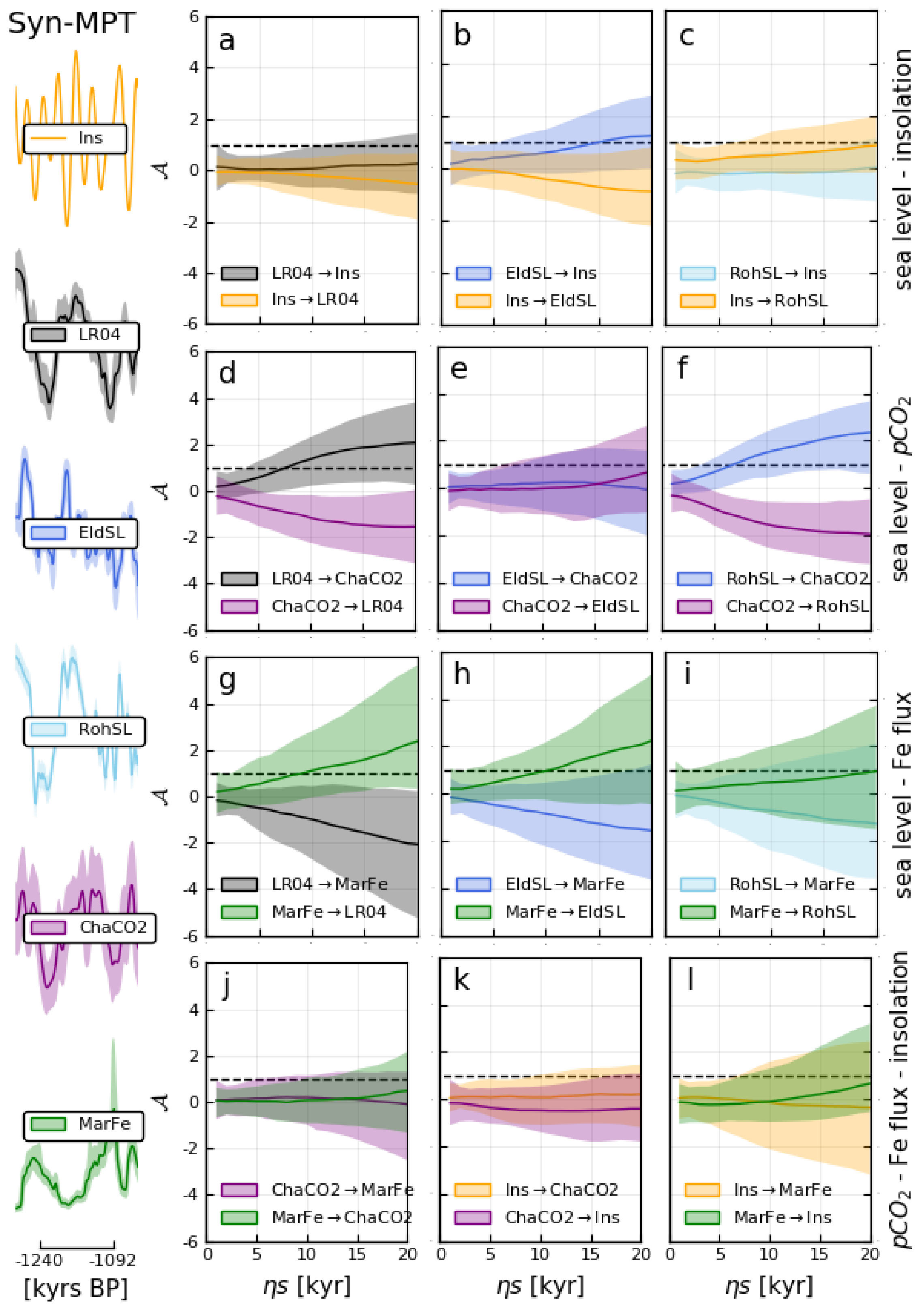
```
p101 = plot(pe_gsl_insol, xlabel = "", xtickfont = false)
p102 = plot(pe_gsl_co2, xlabel = "", xtickfont = false)
p103 = plot(pe_gsl_dust, xlabel = "", xtickfont = false)
    pe_fit3
    #pe_co2_dust,
    #pe_insol_co2,
    #pe_insol_dust,


pe_allresults =
plot(p101, p102, p103, pe_fit3,
    layout = grid(4,1), #size = (750, 4*250),
    ylims = (-6,6), yticks = (-10:2:10),
    size = (72*8.27, 72*11.79) # A4 size
)

for i in 1:12
    annotate!(2,5, panel_labels[i], subplot = i)
end


pe_allresults
savefig("../../results_ePalus_ns20/e_allresults/synMPT_Cgrid_e4_La2004nB_wPanell
abels_A4_.pdf")
```


a
b
c
d
e
f
g
h
i
j
k
l
LR04 → Ins
Ins → LR04
EldSL → Ins
Ins → EldSL
RohSL → Ins
Ins → RohSL
LR04 → ChaCO2
ChaCO2 → LR04
EldSL → ChaCO2
ChaCO2 → EldSL
RohSL → ChaCO2
ChaCO2 → RohSL
LR04 → MarFe
MarFe → LR04
EldSL → MarFe
MarFe → EldSL
RohSL → MarFe
MarFe → RohSL
ChaCO2 → MarFe
MarFe → ChaCO2
Ins → ChaCO2
ChaCO2 → Ins
Ins → MarFe
MarFe → Ins
$\mathcal{A}$
ηs [kyr]
sea level - insolation
sea level - pCO_2
sea level - Fe flux
pCO_2 - Fe flux - insolation
6
4
2
0
-2
-4
-6
0
5
10
15
20

Make an appendix plot'

In [65]:

```
# overview time series

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

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

po_alltimeseries = plot(po0,po1,po3,po4,po6,po8, p_xaxis,
    layout = l, #grid(7,1),
    size = (250, 600),
    xlims = (gridstart, gridend), xticks = (gridstart : gridend-gridstart  : gri
dend),
    ylabel = "", ytickfont = false, yaxis = :off,
    legend = :left, bg_legend = :white)

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


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

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


Syn-MPT
Ins
LR04
EldSL
RohSL
ChaCO2
MarFe
-1240
-1092
[kyrs BP]
a
b
c
d
e
f
g
h
i
j
k
l
𝒜
LR04 → Ins
Ins → LR04
EldSL → Ins
Ins → EldSL
RohSL → Ins
Ins → RohSL
LR04 → ChaCO2
ChaCO2 → LR04
EldSL → ChaCO2
ChaCO2 → EldSL
RohSL → ChaCO2
ChaCO2 → RohSL
LR04 → MarFe
MarFe → LR04
EldSL → MarFe
MarFe → EldSL
RohSL → MarFe
MarFe → RohSL
ChaCO2 → MarFe
MarFe → ChaCO2
Ins → ChaCO2
ChaCO2 → Ins
Ins → MarFe
MarFe → Ins
ηs [kyr]
sea level - insolation
sea level - pCO₂
sea level - Fe flux
pCO₂ - Fe flux - insolation

```
In [ ]:
```

```
In [ ]:
```