NB1: Preparing the data for analysis

Written by Maria Salem, in supplement to master thesis.

In this notebook we prepare all the datasets for analysis by reading in the data, interpolating values where necessary, redefining the data as probability distributions (KDE), and resample the data on a regular time grid.

At the end of the notebook, we prepare any relevant figures to include in the main text data chapter.

The records that will be used in this project are:

1. $\delta^{18}O$

- *Lisiecki & Raymo (2005)* (denoted **LR04**). $\delta^{18}O$ global reference stack, spanning 5.3 Ma

1. Global sea-level (GSL)

- *Spratt & Lisiecki (2016)* (denoted **SL**). GSL stack, spanning 798 ka (PCA)
- *Elderfield et al. (2012)* (denoted **E**). GSL record, spanning 1.5 Ma (Mg/Ca temperature deconvolution of d18O)
- *Grant et al. (2014)* (denoted **G**). Red Sea RSL record, spanning 500? ka.
- *Rohling et al. (2014)* (**R**). Mediterranean RSL record, spanning 5.3 Ma

1. insolation

- *Laskar (2004)* (denoted **La2004**). Numerical solution for northern hemisphere summer insolation (top of atmosphere solar flux for 21^{st} of June at 65°N). 50 Myr back and forth in time from present without significant uncertainty associated (uncertain beyond that).

1. pCO2

- *Bereiter et al. (2015)* (denoted **B**). pCO2 record from ice core at Epica Dome C, spanning ca 800 ka.
- *Chalk et al. (2017)* (denoted **C**). pCO2 record from $\delta^{11}B$ proxy, spanning ca 1.090-1.240 Ma.

1. Dust

- *Lambert et al (2008)* (denoted **L**). Dust record from dust concentration in ice core at Epica Dome C, spanning ca 800 ka. (Note, the data has been set to updated age model since publication.)
- *Martinez-Garcia et al. (2011)* (denoted **MG**). Record of iron mass accumulation rate (Fe MAR) in the Southern Ocean, spanning the last 4 Myr.

Outline for each dataset

- i) **Read in data.** Reference, read in and plot the data for a first overview.
- ii) **Subtle redefinition from age to time.** We reverse the rows `rev` to get arrays where time increases forward with index value (instead of age, increasing backwards). We then name the arrays, and make any necessary adjustments on units ets (f.ex. reformulation from years to kyrs (dividing by 1000). For one of the datasets (Lambert), we also redefine the age model.
- iii) **Additional information on age model and uncertainties** Since our method operates with a lead-lag definition of causality, it is very sensitive to age reversals, and it is crucial to include uncertainties in the time series' age models. For records where age model uncertainties are not included in the dataset, we discuss and include these. For the Lambert record, we also redefine the age model.
- iv) **Discuss time series resolution, and interpolate data if needed**. We will run our general analyses with a timestep of 1000 years (binsize = 1 kyr). Additionally, we **discuss which time series may be suitable for higher resolution analyses.** Since our method requires continuous records, we have to interpolate values for the datasets that have gaps in resolution that are larger than the resolution we will run our analysis on. This is to ensure we won't have any empty bins in the grid we will define later for our analysis (step *vi*). We use linear interpolation, to make as few assumptions as possible. The `LinearInterpolation` function from the *Interpolations* package makes a continuous function of the data where you can pick any time point to carry on. The arrays of interpolated values are marked with the prefix *intp_*.
- v) **Redefine the datasets as `UncertainIndexValueDatasets` (uivDs)**. This datatype from the *UncertainData* package carries on uncertainties in kernel density estimates (KDE). Resampling can then be done from the KDE, which is much faster and more accurate than running computationally heavy resampling loops.
- vi) **Binned resampling of the time series**. Having the time series on the exact same timesteps is crucial for our method to work. We therefore define a regular time grid onto which we bin the datapoints by resampling. This shuffles the uncertainty in age over to the value uncertainty / probability distribution. The `BinnedResampling` function resamples values *from the KDE*, and assigns one value in each bin (each timestep). We choose a binsize (time step) of 1000 years (1 kyr), in accordance with the Spratt & Lisiecki grid, and also this being roughly the order of resolution of out other records. For records with higher resolution, we may choose a finer binsize to run high resolution analyses. Which records are suitable for high resolution analysis and with what time step will be discussed below.

(Note: The *common time grid for analysis* is defined not only by the resolution but also by the common time interval of the time series pairs we want to analyze. Since the BinnedResampling function is computationally heavy and time consuming, we want to avoid running it again for each time series pair we want to analyse. We therefore decide on what will be the common *resolution* and do a binned resampling of the full length of each time series in this notebook. The common *time interval* will be selected later, and define the notebooks for analyis and results (*NBRs*). This saves us a lot of computational time, given that we have so many time series combinations and time intervals to run analyses over.

- vii) **Save the wrangled data**. The the binned resampled uivD time series are saved in a .jld2 file, so that we can smoothly load it in to the next notebooks where we run our analysis, without having to rerun the computations.

(Outline for our next notebooks)

> For overview. Include here or in main text?
>
> NB. OLD. Some repetition since I later moved BinnedResampling from NBRs to NB1.

The next steps for the wrangled data will be to get the time series on a *common time grid*. This is crucial for our method to work. We will therefore define a time grid onto which we bin the datapoints. The time grid is defined according to the length and resolution of the time series pairs we want to analyze. Once a common grid is defined, the time series will be binned on the grid by assigning a corresponding value to each bin. For data with associated uncertainty (uncertainties are carried in the uivD-objects), binning the data to the time grid is done by the function `BinnedResampling` . This function resamples the values within each bin. This generates a probaliblity distribution, allowing us to quantify the confidence for the mean value assigned to each bin. We will represent all our data with a 95% confidence envelope, for robust results.

Once the datasets are binned to a common time grid, we can run our analyses.

Note: The different time series combinations will require different common grids. We will therefore create different notebooks (NBRs) for the different common grids. These are the notebooks where we will compute our results (**NBR**, see below).

NB2

In notebook 2, we thoroughly go through all the steps in our analysis. We use the example of an autoregressive system of the first order, from which we create two time series realizations X and Y, and show how Predictive Asymmetry is estimated between timeseries data. The notebook walks through all the steps in the analysis, with abundant comments and explanatory figures. The aim is to give a general understanding of predictive asymmetry and of what is done in the following notebooks (NB3 and NBRs).

(After synthetic time series are represented, include note on reflections/challenges for using real-world empirical time series. -introduce binned resampling here instead?)

NB3

In notebook 3, we summarize the the code in notebook 2 into functions. This allows us to run the analysis between the many time series more efficiently.

NBRs

The results notebooks (NBR) is where we run the analyses between our time series data and plot our results. We include the functions for analysis defined in NB3, and the time series prepared in this notebook (NB1). We make a separate NBR for every grid (time interval) we will run analysis over. Each time grid is constrained primarily by the timespan and resolution of the available records. (ice core records, for example, only cover the last 800 kyrs). We also define time span constraints on time grids that allow us to check if dynamical coupling differs over the hypothesized periods post-MPT (0-800 kyr BP), syn-MPT (800-1250 kyr BP) and pre_MPT ((1250-4000 kyr BP).

Loading in the time series data and wrangle data

First, we load in the necessary packages/libraries

In [2]:

```
# loading in the necessary packages/libraries

using
Pkg, # For adding the libraries we will use
Distributions, # for probability distributions and associated functions
DelimitedFiles,  # For reading in dataset files that are .tab or .csv or .txt
Plots; # for making the figures
pyplot() # running pyplot in backend to plots (not as pretty as GR, but faster,
 and compatible with LaTeXStrings)

using
LaTeXStrings, # For writing special symbols and characters
JLD2, # For saving the arrays we want to carry on in different notebooks
DataInterpolations # For linear interpolations on time series with discontinuous
or insufficient resolution

using
UncertainData, # for carrying on uncertainties as kernel density estimates (type
UncertainIndexValueDatasets), and BinnedResampling
# maybe not used here?
CausalityTools, # ...???
DynamicalSystems # ...??? A Julia suite for chaos and nonlinear dynamics


# not sure if I use these at all?
using
StatsBase, # ???
Test, # ???
XLSX # For reading in dataset files written in excel
;
```

In [2]:

```
# if kernel dies repeatedly, then try rebuilding
Pkg.build("IJulia")
```

```
  Building Conda ──→ `~/.julia/packages/Conda/3rPhK/deps/build.log`
  Building ZMQ ───→ `~/.julia/packages/ZMQ/ItfqT/deps/build.log`
  Building MbedTLS → `~/.julia/packages/MbedTLS/a1JFn/deps/build.log
`
  Building IJulia ─→ `~/.julia/packages/IJulia/F1GUo/deps/build.log`
```

Out[2]:

```
false
```

1. LR04

- Lisiecki & Raymo (2005), denoted **LR04**
- Global $\delta^{18}O_b$ stack, spanning the last 5.32 Myrs.
- Both the LR04 age model and $\delta^{18}O$ serve as reference stacks in paleoclimatologic sciences.
- Data available from Pangaea, DOI: https://doi.org/10.1594/PANGAEA.704257 (https://doi.org/10.1594/PANGAEA.704257)
- *NB: the LR04 age model is constructed using orbital tuning. This may create bias in our results in analyses between this record and insolation time series.*

i) Read in the data

In [33]:

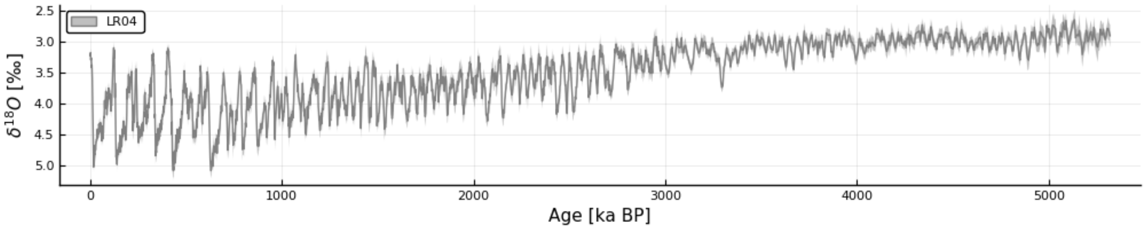
```
# Read in dataset from Lisiecki & Raymo

filepath_LR04 = "../../MASTER_2.0/data/sea-level/LR04/Lisiecki_Raymo_2005/datase
ts/Global_stack_d18O.tab"
rawD_LR04 = readdlm(filepath_LR04, skipstart = 58)
;
```

In [34]:

```
# name columns
age_LR04_ = rawD_LR04[:,1]       # Age [ka BP]
d18O_LR04_ = rawD_LR04[:, 2]    # δ18O [‰]
d18O_1σ_LRO4_ = rawD_LR04[:,3] # standard deviation [±]
;
```

In [36]:

```
# plot LR04 record
plot_LR04_raw_age =
plot(age_LR04_, d18O_LR04_,
    label = "LR04", color = :gray,
    ribbon = 2*d18O_1σ_LRO4_,
    yflip = true, # by convention, we plot d18O on a reversed axis #(mirror read
ability of GSL, which constitutes on average 70% of the signal)
    xlabel = "Age [ka BP]",
    ylabel = L"\delta^{18}O  \ [\perthousand]",
    size = (1000,200))

savefig("../../MASTER_2.0/figurar/RawData/LR04_age.pdf")
```



ii) **Subtle redefinition of record as time series.** We reverse dataset to go from age, increasing backwards, to time, increasing forwards. To keep sensical index values, e define present as 0 kyrs BP.

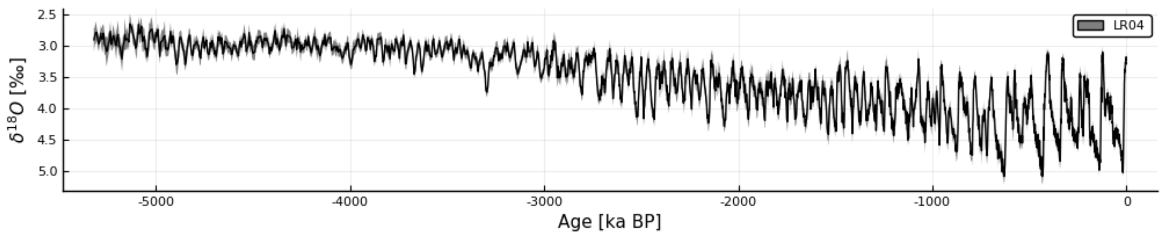
In [37]:

```
# Reverse dataset to go from age increasing backwards to time increasing forward
s.
revD_LR04 = reverse(rawD_LR04, dims = 1)

# name columns
t_LR04 = -revD_LR04[:,1]      # Age [ka BP] # negative, to define present as 0 k
yrs
d18O_LR04 = revD_LR04[:, 2]   # δ18O [‰]
d18O_1σ_LRO4 = revD_LR04[:,3] # standard deviation [±]
;
```

In [38]:

```
# plot LR04 time series
plot_LR04_raw_time =
plot(t_LR04, d18O_LR04,
    label = "LR04",
    color = :black,
    ribbon = 2*d18O_1σ_LRO4,
    xlabel = "Age [ka BP]",
    ylabel = L"\delta^{18}O  \ [\perthousand]",
    yflip = true, # by convention, we plot d18O on a reversed axis #(mirror read
ability of GSL, which constitutes on average 70% of the signal)
    size = (1000,200)
    )

#savefig("../figurar/RawData/LR04_time.pdf")
```

Out[38]:



iii) **Potential systematic uncertainties in age model**

Since our method operates with a lead-lag definition of causality, it is very sensitive to age reversals, and it is crucial to include uncertainties in the time series' age models.

Lisiecki & Raymo (2005) report the LR04 age model uncertainties as following:

"Including all sources of error, we estimate the uncertainty in the LR04 age model to be 40 ky from 5.3-5 Ma, 30 ky from 5-4 Ma, 15 ky from 4-3 Ma, 6 ky from 3-1 Ma, and 4 ky from 1-0 Ma".

> Kommentar frå Bjarte: kanskje overdreven usikkerhet. Men me bruker som dei oppgir i første omgang (konservativt), og kan evt kjøre sensitivitetstest seinare med mindre usikkerhet.

We interpret this as a 95% confidence interval ($\pm 2\sigma$), and define an array to carry on the age uncertainty:

In [39]:

```
# Make an array of the potential systematic deviations in the LR04 age model

t_4σ_LR04 = zeros(length(t_LR04))

t_4σ_LR04[ t_LR04 .> -1000] .= 4                          # 4 ky from -1 Ma to 0
 (present)
t_4σ_LR04[(t_LR04 .<= -1000) .& (t_LR04 .> -3000)] .= 6  # 6 ky from -3 to -1 Ma
t_4σ_LR04[(t_LR04 .<= -3000) .& (t_LR04 .> -4000)] .= 15 # 6 ky from -4 to -3 Ma
t_4σ_LR04[(t_LR04 .<= -4000) .& (t_LR04 .> -5000)] .= 30 # 6 ky from -3 to -1 Ma
t_4σ_LR04[ t_LR04 .<= -5000] .= 40                        # 40 ky for before -5 M
a

t_1σ_LR04 = t_4σ_LR04 ./ 4
;
```

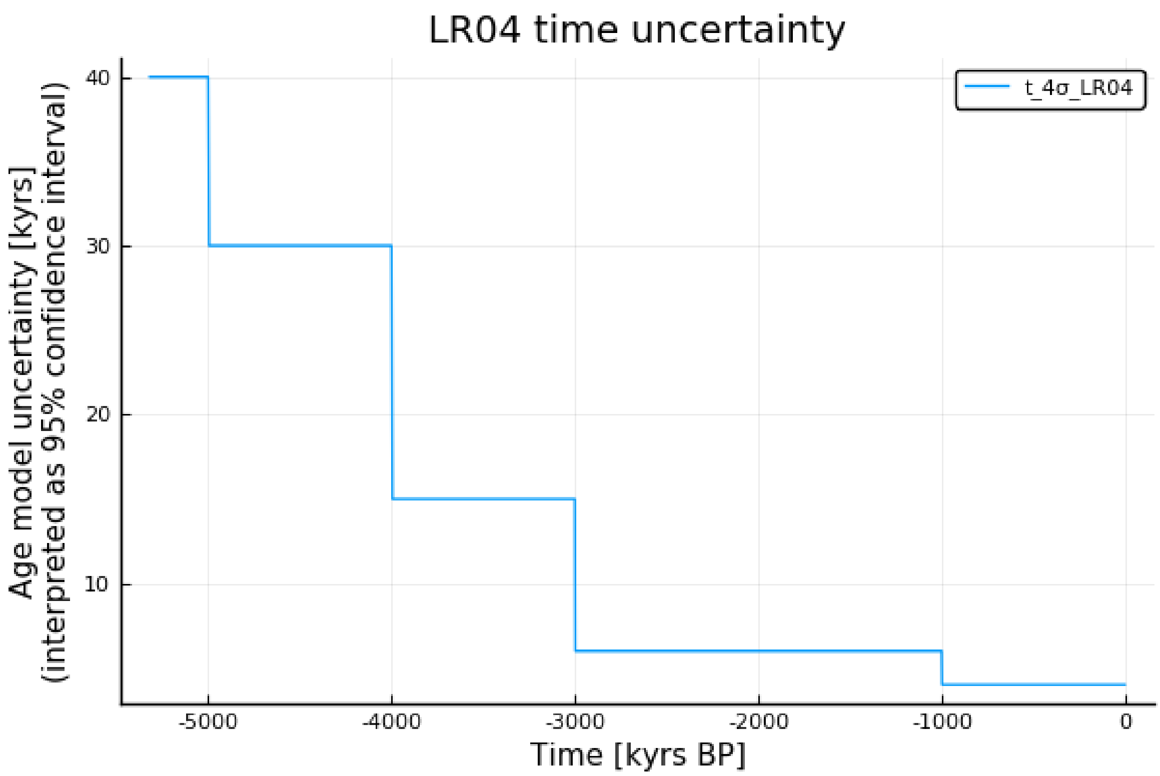
In [40]:

```
# we now have all the arrays we need to produce the raw plot, lets save them in
 a .jld2 file for easy access
@save "../Koding/WrangledDataFiles/BasicArrays/LR04.jld2" t_LR04 t_1σ_LR04 d18O_
LR04 d18O_1σ_LRO4
```

In [45]:

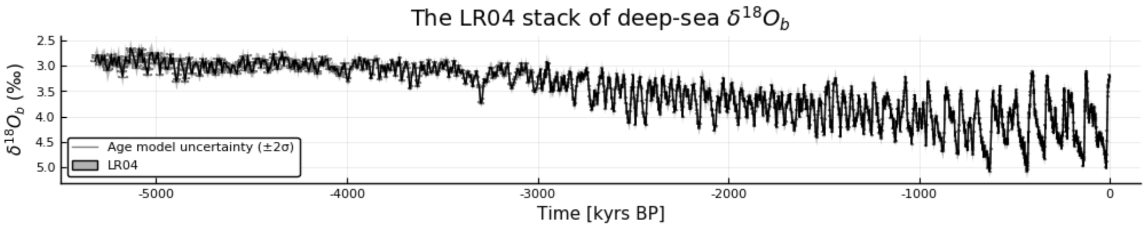
```
plot(title = "LR04 time uncertainty",
    t_LR04, t_4σ_LR04, # Showing stepwise definition of age uncertainty, where u
ncertainty increases back in time. All  good.
    xlabel = "Time [kyrs BP]",
    ylabel = "Age model uncertainty [kyrs]
(interpreted as 95% confidence interval)",
    label = "t_4σ_LR04")
```

Out[45]:



In [51]:

```
# plot the LR04 time series
plot_LR04_raw_timeunc =
plot(title = string("The LR04 stack of deep-sea ", L"$\delta^{18}O_{b}$"),
    label = "LR04",
    size = (1000,200),
    xlabel = "Time [kyrs BP]",
    ylabel =  L"\delta^{18}O_{b} \ (\perthousand)",
    yflip = true # by convention, we plot d18O on a reversed axis #(mirror reada
bility of GSL, which constitutes on average 70% of the signal)
    )
plot!(t_LR04, d18O_LR04,
    xerr = (2*t_1σ_LR04, 2*t_1σ_LR04), # age uncertainty, what we have interpret
ed as 95% confidence interval
    ms=1, color = :grey,
    label = "Age model uncertainty (±2σ)"
    )
plot!(t_LR04, d18O_LR04,
    ribbon = (2*d18O_1σ_LRO4, 2*d18O_1σ_LRO4), # plotting with 2σ (95% confidence
interval)
    color = :black, fillalpha = 0.3,
    label = "LR04"
    )

savefig("../../Master_2.0/figurar/RawData/LR04_timeunc.pdf") # NB currently save
d with the plot!(xerr)
```



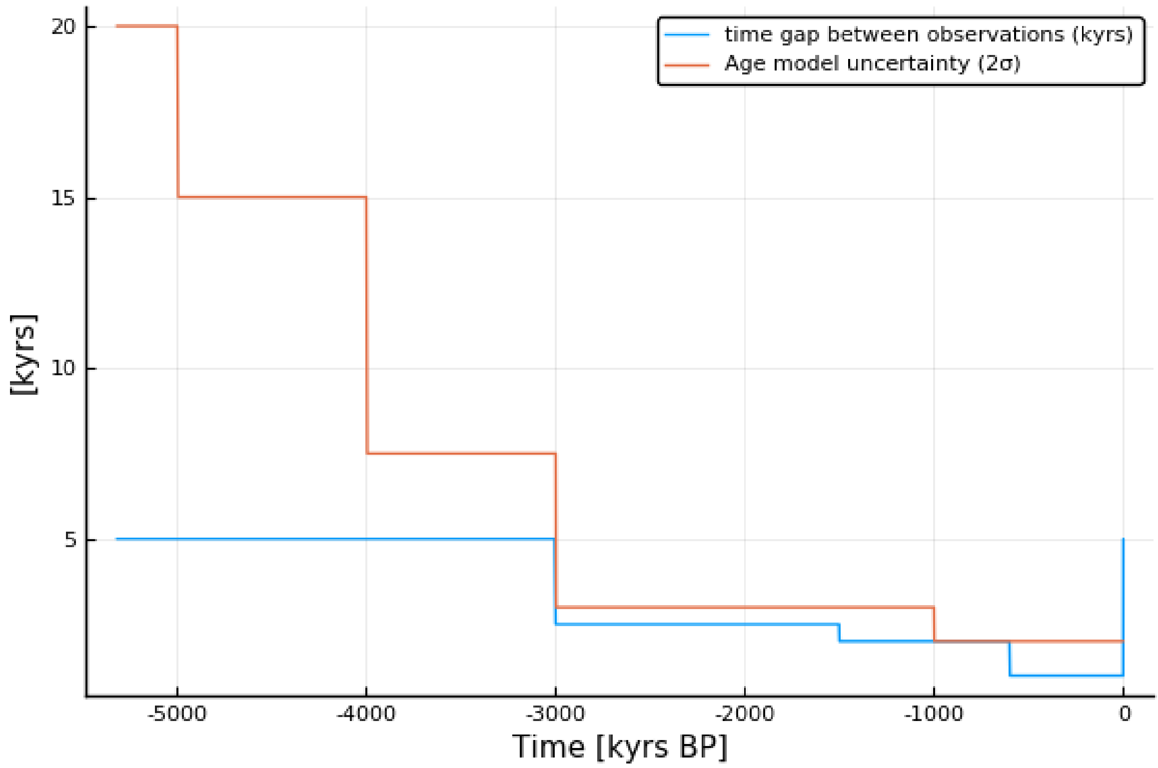
Interpolation. In the final analysis we use a 1000 year resolution grid (defined by SprattLisiecki grid), and we cannot have any empty bins in our final analysis. We therefore need to check that our data has at least 1000 years resolution, and if not, we need to interpolate values where there is missing information.

In [54]:

```
# Check if need for interpolation?

print("Smallest time gap between observations is ", minimum(diff(t_LR04)), " kyr
s.
")
print("Mean time gap between observations is ", mean(diff(t_LR04)), " kyrs.
")
print("Largest time gap between observations is ", maximum(diff(t_LR04)), " kyr
s.
")

plot(t_LR04, diff(t_LR04),
xlabel = "Time [kyrs BP]",
ylabel = "[kyrs]",
label = "time gap between observations (kyrs)")
plot!(t_LR04, 2*t_1σ_LR04, label = "Age model uncertainty (2σ)")
```

```
Smallest time gap between observations is 1.0 kyrs.
Mean time gap between observations is 2.5165562913907285 kyrs.
Largest time gap between observations is 5.0 kyrs.
```

Out[54]:




```
WARNING: both StateSpaceReconstruction and DynamicalSystems export
"dimension"; uses of it in module CausalityTools must be qualified
```

t_LR04 # ends at present (0 yrs BP), with 1 kyr between datapoints. So why the big gap at the end here?

In [103]:

```
# Question about the plot
t_LR04 ;# ends at present (0 yrs BP), with 1 kyr . So why the big gap at the end
here?
```

Based on the ensemble of records we want to analyse, we have chosen to use millennial resolution as default for our analysis.

As visualized in the plot above, the LR04 record has segments with varying resolutions (each segment on different regular grid). In about half the record, there is up to 5 kyrs between each datapoint in the LR04 record. Only in the verymost recent part of the record do we have millenial resolution.

> By including the age model uncertainty (which is larger than the time gaps of the record), the binned resampling will still assign one value to every bin of the 1 kyr grid.
>
> If we choose to not include the age model uncertaint, we will need to interpolate values for the LR04 record. To make the least possible assumptions, we use linear interpolation between the datapoints in the record.

In [9]:

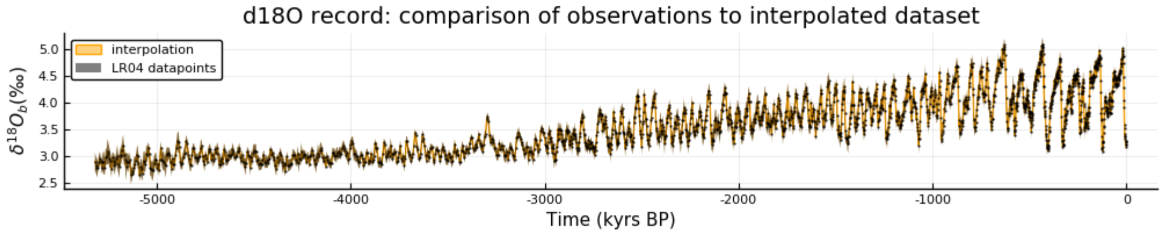
```
# interpolation

# create a continuous function with linear interpolation between every datapoint
in the array.
# This allows us to choose the interpolated value for any point in time.
interpolate_t_LR04 = LinearInterpolation(t_LR04, t_LR04)            # time array
interpolate_mean_LR04 = LinearInterpolation(d18O_LR04, t_LR04)      # mean d18O v
alue
interpolate_1σ_LR04 = LinearInterpolation(d18O_1σ_LRO4, t_LR04)     # d18O value
 uncertainties
interpolate_t_1σ_LR04 = LinearInterpolation(t_1σ_LR04, t_LR04)      # age model u
ncertainty (full)
#interpolate_t_1σ_LR04_trunc = LinearInterpolation(t_1σ_LR04_trunc, t_LR04) # ag
e model uncertainty, truncated at 1 sigma

# make a fine grained grid to contain the interpolated values
binsize_intp = 0.1
fine_grid_LR04 = ceil(minimum(t_LR04)) : binsize_intp : floor(maximum(t_LR04)) #
One bin for every 100 years (0.1 kyrs)
print(fine_grid_LR04)

# make new arrays for interpolated data
# give a value from the interpolate function to every bin in the fine grid
intpD_t_LR04 = [interpolate_t_LR04(i) for i in fine_grid_LR04]
intpD_mean_LR04 = [interpolate_mean_LR04(i) for i in fine_grid_LR04]
intpD_1σ_LR04 = [interpolate_1σ_LR04(i) for i in fine_grid_LR04]
intpD_t_1σ_LR04 = [interpolate_t_1σ_LR04(i) for i in fine_grid_LR04]

#intpD_t_1σ_LR04_trunc = [interpolate_t_1σ_LR04_trunc(i) for i in fine_grid_LR0
4] # Error here, so we truncate below
#intpD_t_1σ_LR04_trunc = Truncated.(Normal.(intpD_t_LR04, intpD_t_1σ_LR04), intp
D_t_LR04 .- intpD_t_1σ_LR04, intpD_t_LR04 .+ intpD_t_1σ_LR04)
;
```

```
-5320.0:0.1:0.0
```

In [105]:

```
# plot to check

plot(title = "d18O record: comparison of observations to interpolated dataset",
    size = (1000,200),
    xlabel = "Time (kyrs BP)",
    ylabel =  L"\delta^{18}O_{b} (\perthousand)")
plot!(intpD_t_LR04, intpD_mean_LR04,
    ribbon = (2 * intpD_1σ_LR04),
    color = :orange,
    ms = 0.1,
    label = "interpolation")
scatter!(t_LR04, d18O_LR04,
    ribbon = (2 * d18O_1σ_LRO4),
    color = :black,
    ms = 0.5,
    label = "LR04 datapoints")


# We see that the original time series and the interpolated time series overlap,
so all good
```

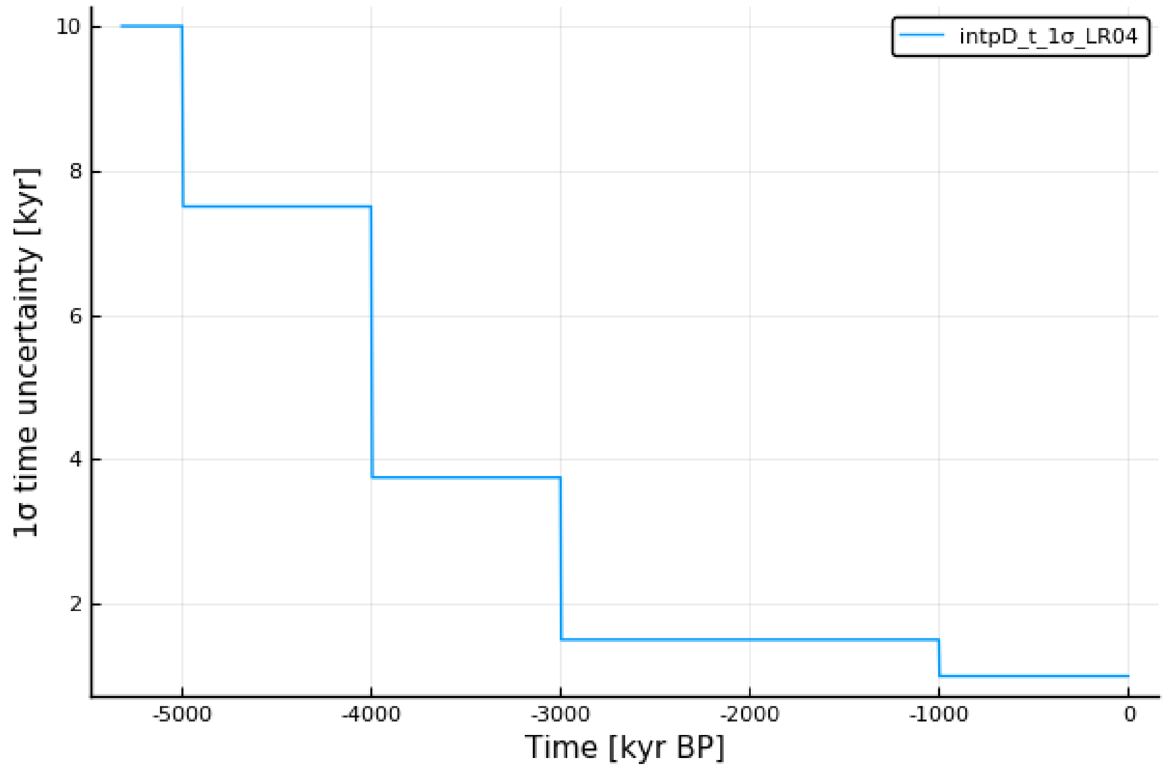
Out[105]:



In [84]:

```
plot(#title = "LR04 time uncertainty - interpolated array",
    intpD_t_LR04, intpD_t_1σ_LR04,
    xlabel = "Time [kyr BP]",
    ylabel = "1σ time uncertainty [kyr]",
    label = "intpD_t_1σ_LR04")

# We see from the figure that the interpolated time uncertainty array is the sam
e as the original one we made above, so all good.
```

Out[84]:



Redefine as an `UncertainIndexValueDataset`

The type object `UncertainIndexValueDataset` from `the UncertainData` package is a format to carry on data with associated uncertainties in one single array. The uncertainties are carried on as kernel density estimates (KDE). This is less computationally heavy than the traditional for-loop resampling. This datatype is seamlessly integrated in the DynamicalSystems and CausalityTools packages, which we will use onwards in the NBRs.

In [55]:

```
# LR04 with maximum age uncertainty

d18O_uiv_LR04 = [UncertainValue(Normal, d18O_LR04[i], d18O_1σ_LRO4[i]) for i in
1:length(d18O_LR04)] ;
t_uiv_LR04_fullageunc = [UncertainValue(Normal, t_LR04[i], t_1σ_LR04[i]) for i i
n 1:length(t_LR04)]
uivD_LR04_fullageunc = UncertainIndexValueDataset(t_uiv_LR04_fullageunc, d18O_ui
v_LR04)
```

Out[55]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 2115 uncertain values coupled with 2115 uncertain ind
ices
```

In [85]:

```
# LR04 with maximum age uncertainty

d18O_uiv_LR04_intp = [UncertainValue(Normal, intpD_mean_LR04[i], intpD_1σ_LR04[i
]) for i in 1:length(intpD_mean_LR04)] ;
t_uiv_LR04_fullageunc_intp = [UncertainValue(Normal, intpD_t_LR04[i], intpD_t_1σ
_LR04[i]) for i in 1:length(intpD_t_LR04)]
uivD_LR04_fullageunc_intp = UncertainIndexValueDataset(t_uiv_LR04_fullageunc, d1
8O_uiv_LR04)
```

Out[85]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 53201 uncertain values coupled with 53201 uncertain i
ndices
```

In [ ]:

```
#plot(uivD_LR04_fullageunc)
```

```
InterruptException:

Stacktrace:
 [1] _update_series_attributes!(::Dict{Symbol,Any}, ::Plots.Plot{Plo
ts.PyPlotBackend}, ::Plots.Subplot{Plots.PyPlotBackend}) at /Users/m
aria/.julia/packages/Plots/qZHsp/src/args.jl:1598
 [2] _process_seriesrecipe(::Plots.Plot{Plots.PyPlotBackend}, ::Dict
{Symbol,Any}) at /Users/maria/.julia/packages/Plots/qZHsp/src/pipeli
ne.jl:403
 [3] _process_seriesrecipe(::Plots.Plot{Plots.PyPlotBackend}, ::Dict
{Symbol,Any}) at /Users/maria/.julia/packages/Plots/qZHsp/src/pipeli
ne.jl:417
 [4] _plot!(::Plots.Plot{Plots.PyPlotBackend}, ::Dict{Symbol,Any},
::Tuple{UncertainIndexValueDataset{UncertainIndexDataset,UncertainVa
lueDataset}}) at /Users/maria/.julia/packages/Plots/qZHsp/src/plot.j
l:234
 [5] #plot#138(::Base.Iterators.Pairs{Union{},Union{},Tuple{},NamedT
uple{(),Tuple{}}}, ::typeof(plot), ::UncertainIndexValueDataset{Unce
rtainIndexDataset,UncertainValueDataset}) at /Users/maria/.julia/pac
kages/Plots/qZHsp/src/plot.jl:57
 [6] plot(::UncertainIndexValueDataset{UncertainIndexDataset,Uncerta
inValueDataset}) at /Users/maria/.julia/packages/Plots/qZHsp/src/plo
t.jl:51
 [7] top-level scope at In[109]:1
```

The Age model uncertainties reported (from tie points, not included in the original dataset) were quite substantial, and may overestimate the uncertainty (...). Therefore, let's make three versions of the record:

- one without age uncertainty,
- one including the 95% confidence interval,
- and one truncated at 1σ.

As a starting point, we will use (truncated? full? none?)

We can later do a sensitivity analysis, to check how the different uncertainty levels (**?**) impact the causality results.

In [56]:

```
# LR04 without age model uncertainty
t_uiv_LR04_noageunc = [UncertainValue(Normal, intpD_t_LR04[i], 0) for i in 1:len
gth(intpD_t_LR04)]
uivD_LR04_noageunc = UncertainIndexValueDataset(t_uiv_LR04_noageunc, d18O_uiv_LR
04)
```

```
UndefVarError: intpD_t_LR04 not defined

Stacktrace:
 [1] top-level scope at In[56]:1
```

> Should we run our analysis with full uncertainty? Or truncate?
>
> Useful to do both, with sensitivity analysis, to check what impact it would have on the results

Eksempel frå Kristian på korleis ta med aldersusikkerhet. normalfordeling = Truncated(Normal(2, 0.2), 2-0.2, 2+0.2) # (normalfordeling(mean, 1σ), trunkert slik at nedre grense, øvre grense = +-1sigma) # rand(normalfordeling) # sjekk at trunkering blir som forventa (2 +/- 0.2)# LR04 with age uncertainty truncated at +/-1σ intpD_t_1σ_LR04_trunc = Truncated.(Normal.(intpD_t_LR04, intpD_t_1σ_LR04), # Trunker(normalfordeling(mean, 1σ) intpD_t_LR04 .- intpD_t_1σ_LR04, # trunkert slik at nedre grense = mean - 1σ intpD_t_LR04 .+ intpD_t_1σ_LR04) # trunkert slik at øvre grense = mean + 1σ length(intpD_t_1σ_LR04_trunc) # 53201 length(intpD_t_LR04) # 53201 typeof(intpD_t_1σ_LR04_trunc) # ...So I don't get why I can't reformulate as uivD below....# Truncated age model uncertainty redefined as an uivD # I wanted to use truncated age uncertainty (truncated at 1σ) for the first round of analysis, but cannot define truncated values as uivD t_uiv_LR04_truncated = [UncertainValue(Normal, intpD_t_LR04[i], intpD_t_1σ_LR04_trunc[i]) for i in 1:length(intpD_t_LR04)] #uivD_LR04_truncated = UncertainIndexValueDataset(t_uiv_LR04_truncated, d18O_uiv_LR04)

> I wanted to use truncated age uncertainty (truncated at 1σ) for the first round of analysis, but DOESN't WORK to define truncated values as uivD
>
> ---

We now have the object in the right format for analysis, an `UncertainIndexValueDataset` that carries on uncertainties in KDE.

In the meanwhile, we make a **.jld2 file to contain the wrangeld data of the LR04 record**:

In [88]:

```
# Save the relevant arrays of the LR04 record in a .jld2 file

uivD_LR04 = uivD_LR04_noageunc
# MERK: uivD_LR04 er no lagra utan aldersusikkerhet, for det gav heilt bananas k
onfidensintervall.
# Har sidan funne ut (Jo sjekka) at aldersusikkerheten ikkje var inkludert, så d
et blir rett å bruke fullageunc

@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/LR04.jld2" uivD_LR04_noag
eunc uivD_LR04_fullageunc #uivD_LR04_truncated
```

In [89]:

```
@load "../Koding/WrangledDataFiles/uivDs/LR04.jld2"
```

Out[89]:

```
2-element Array{Symbol,1}:
 :uivD_LR04_noageunc
 :uivD_LR04_fullageunc
```

In [ ]:

```
#plot(uivD_LR04)
```

vi) Binned Resampling

The next step will be to perform BinnedResampling to get on a common time grid to run our analyses. What the common grid is will vary, depending on the length and resolution of the time series it is paired with for analysis. The definition of the common grid and ensuing BinnedResampling of the time series will therefore be done in the notebooks for analyses and results (NBRs), where every every NBR is defined by a different grid.

Define a grid for BinnedResampling method to allow one value for every 1000 years. It is important in analyses that all values are on the same time grid. For the initial analysis we will use a time step of 1000 years (following the example of the Spratt&Lisiecki sea level record). We may later experiment with higher and lower resolution (smaller and larger binsizes) in sensitivity analyses.

In [81]:

```
binsize = 1 # time step
```

Out[81]:

```
1
```

Intuitively, the grid would run between the first to the last point in the time series. However, since we want the datapoints (bin median) to fall on every discrete 1000 years, like in the Spratt Lisiecki grid, we must ensure that the bin midpoint falls on an integer. we therefore let the bin edges be midways between the intergers (every 0.5, 1.5, 2.5 etc kyrs). To avoid skewedness in the outermost bins or need for extrapolation, we sacrifice the time series length by a datapoint in each end (we start the grid half a binsize *over* the first point, rather than below, and end the grid hald a binsize *before* the last datapoint).

In [83]:

```
#= Define a grid for binned resampling,
with approx the length of LR04 record, and one bin for every 1000 years.
Formulate as to assign one value at every discrete 1 kyr (bin midpoint at Intege
rs)) =#

tmin_LR04 = ceil(minimum(t_LR04)) # -5320 kyrs BP
tmax_LR04 = floor(maximum(t_LR04)) # 0 kyrs BP (present)
binmidpoints_LR04 = tmin_LR04 : binsize : tmax_LR04
grid_LR04 = tmin_LR04 - binsize/2 : binsize : tmax_LR04 + binsize/2
```

Out[83]:

```
-5320.5:1.0:0.5
```

- Defining the BinnedResampling method:

In [59]:

```
#= Define the resampling method:
in each bin of the grid_LR04, draw 1000 resamples (with substitution) =#
resampling_method_LR04 = BinnedResampling(grid_LR04, 1000)
```

Out[59]:

```
BinnedResampling{StepRangeLen{Float64,Base.TwicePrecision{Float64},B
ase.TwicePrecision{Float64}}}(-5320.5:1.0:0.5, 1000)
```

- Compute a binned resampling of the LR04 record on the defined 1 kyr grid, drawing a 1000 resamples in each bin.

In [68]:

```
@time LR04_binned_fullength_fullageunc = resample(uivD_LR04_fullageunc, resampli
ng_method_LR04)
```

```
 77.944174 seconds (45.99 M allocations: 48.826 GiB, 39.04% gc time)
```

Out[68]:

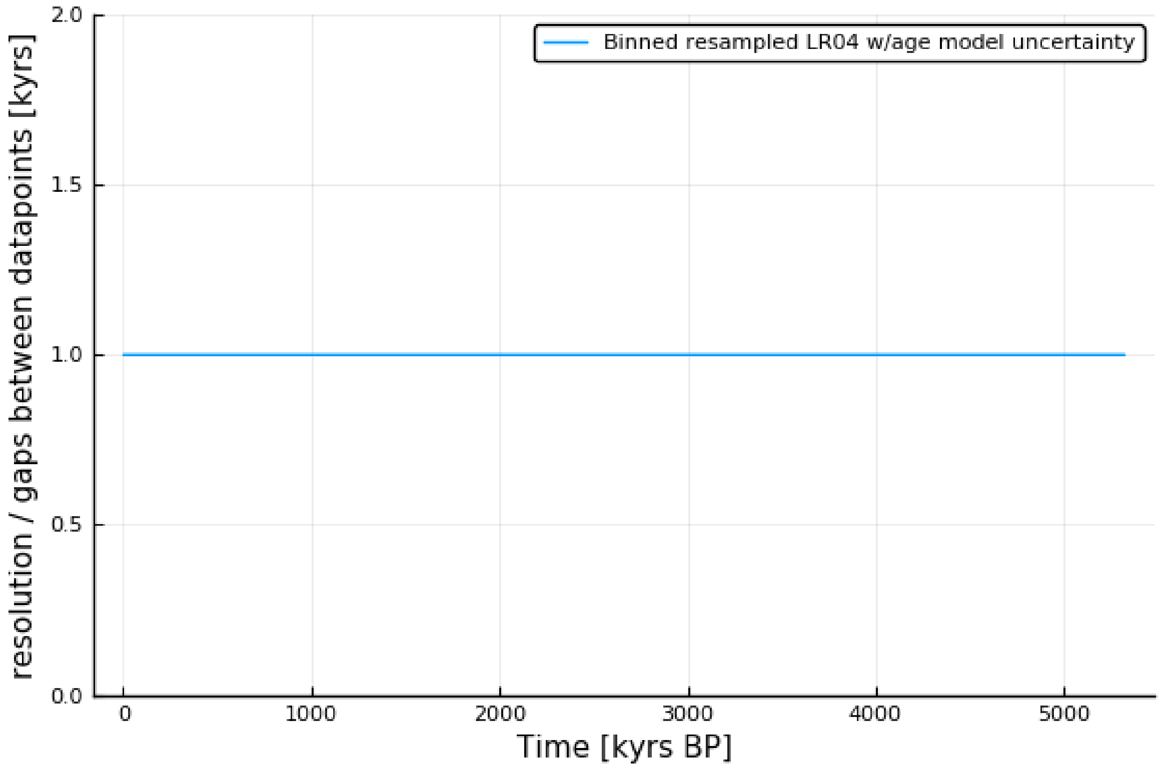
```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 5321 uncertain values coupled with 5321 uncertain ind
ices
```

- Check that we have a value in all bins

In [107]:

```
# check that we have a value in all bins
binmidpoints = [LR04_binned_fullength_fullageunc.indices[i].value for i in 1:len
gth(LR04_binned_fullength_fullageunc)]
plot(diff(binmidpoints),
    xlabel = "Time [kyrs BP]",
    ylabel = "resolution / gaps between datapoints [kyrs]",
    label = "Binned resampled LR04 w/age model uncertainty",
    ylims = (0,2),)


# the record is on a regular grid, all good.
```

Out[107]:



In [93]:

```
LR04_binned_fullength_noageunc_intp = resample(uivD_LR04_noageunc, resampling_me
thod_LR04)
#LR04_binned_truncatedageunc = resample(uivD_LR04_truncated, resampling_method_L
R04)
```

```
UndefVarError: uivD_LR04_noageunc not defined

Stacktrace:
 [1] top-level scope at In[93]:1
```

In [69]:

```
@save "../Koding/WrangledDataFiles/Binned_ts_fullength/LR04.jld2"  LR04_binned_f
ullength_fullageunc LR04_binned_fullength_noageunc_intp
```

Plots

Plot the binned resampled uivD LR04 time series. We will plot the 95% confidence interval, for a solid communication of the uncertainty.

In [67]:

```
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/LR04.jld2"
```

Out[67]:

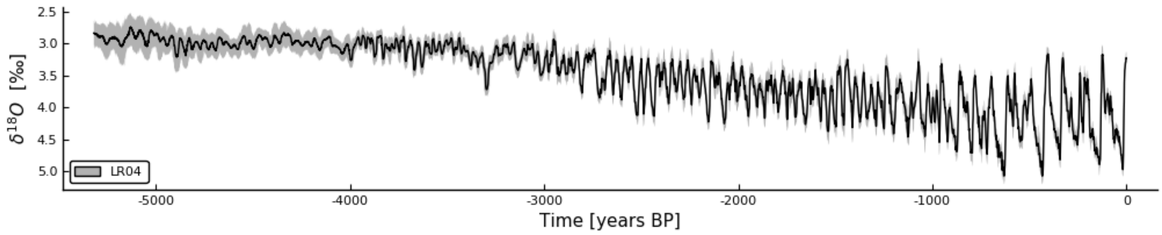
```
2-element Array{Symbol,1}:
 :LR04_binned_fullength_fullageunc
 :LR04_binned_fullength_noageunc_intp
```

1. with age model uncertainty:

In [78]:

```
### Plot the binned resampled uivD LR04 time series with the 95% confidence inte
rval
LR04 = LR04_binned_fullength_fullageunc

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(LR04.values, 0.5)
bin_upper = quantile.(LR04.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(LR04.values, 0.025)
;

plot_LR04_fullageunc =
plot(binmidpoints_LR04, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :black, fillalpha = 0.3,
    label = "LR04",
    xlabel = "Time [years BP]",
    ylabel = L"$\delta^{18}O$  $[\perthousand]$",
    grid = false, yflip = true,
    size = (1000,200), legend = :bottomleft
    )
```

Out[78]:

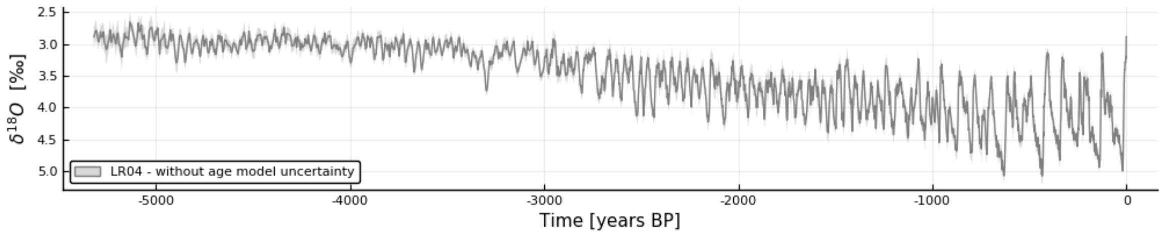


1. without taking into account the age model uncertainty:

In [80]:

```
### Plot the binned resampled uivD LR04 time series with the 95% confidence inte
rval
LR04 = LR04_binned_fullength_noageunc_intp

# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(LR04.values, 0.5)
bin_upper = quantile.(LR04.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(LR04.values, 0.025)
;

plot_LR04_noageunc =
plot(binmidpoints_LR04, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :grey, fillalpha = 0.3,
    label = "LR04 - without age model uncertainty",
    xlabel = "Time [years BP]",
    ylabel = L"$\delta^{18}O$  $[\perthousand]$",
    yflip = true, #grid = false,
    size = (1000,200), legend = :bottomleft
    )
```

Out[80]:



2. Global sea level (GSL)

Due to different uncertainties/methodologies and limited overlap between the sea level time series, we operate with 4 different sea-level records to increase the robustness of our analysis:

- SL = Spratt & Lisiecki (2016) - PCA of 5 different sea-level records spanning the last 800 ka
- E = Elderfield et al. (2012) - Temperature deconvolution of $\delta^{18}O_b$ - spanning 1.5 Ma, tuned to LR04
- G = Grant et al. (2014) - Red Sea $\delta^{18}O_b$ stack, tuned to Sanbao speleothem by record of common monsoon-signal.
- R = Rohling et al. (2014) - Mediterranean $\delta^{18}O_b$ stack, tuned to Sanbao speleothem by record of common monsoon-signal.

2.1 - Spratt-Lisiecki GSL stack

- GSL stack from *Spratt & Lisiecki (2016)*, denoted SL
- Principal component analysis (PCA) of a stack of 5 GLS records, spanning the last 798 kyr.
- Note 1: the SL stack is tuned to the LR04, which is orbitally tuned. This may create bias in our results in analyses between this record and insolation time series.
- Note 2: no uncertainty communicated in Pangaea dataset. Have therefore used dataset from Jo.

> **Check with Jo where data comes from.** (Article reads 1σ "Bootstrapping and random sampling yield mean uncertainty estimates of 9-12 m (1σ)").

i) Read in dataset from Spratt & Lisiecki

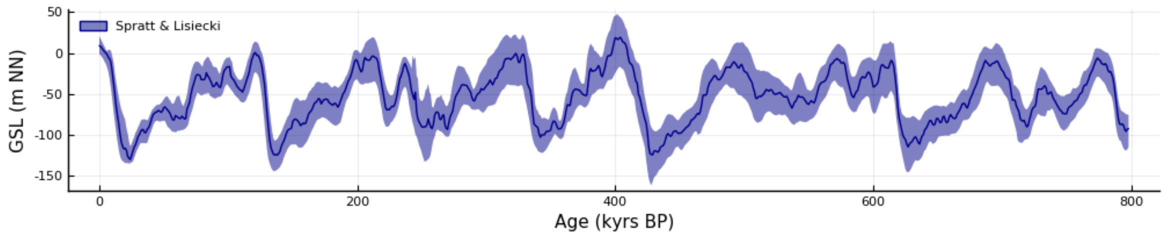
In [51]:

```
# Read in dataset from Spratt & Lisiecki

filepath_SL = "../../MASTER_2.0/data/sea-level/SprattLisiecki/SL_v4(Jo).txt"
rawD_SL = DelimitedFiles.readdlm(filepath_SL,'\t', Float64, '\r', dims = (799, 7
), skipstart = 1)

# Naming the arrays
t_SL_ = rawD_SL[:,1]/1000     # age (kyrs), (converted from years to kyrs by div
iding with 1000.)
GSL_mean_SL_  = rawD_SL[:,2]   # SeaLev_longPC1 # mean sea level (m)
GSL_1σ_SL_  = rawD_SL[:,3]     # sea level uncertainty 1\sigma
GSL_SL_err_lo_  = rawD_SL[:,4] # lower 95% confidence interval quantile from Mont
e Carlo analysis
GSL_SL_err_up_  = rawD_SL[:,5] #  upper 95% CI quantile from Monte Carlo analysis

# SprattLisiecki is tuned to LR04 (meaning it has relicts of orbital tuning)
# Jo has tuned the stack to the Sanbao speleothem, to have the GSL stack on the
 same age model as the Mediterranean record (Rohling)
## SHOULD WE USE Jo's AGE MODEL?
SL_t_speleotuning = rawD_SL[:,6]
SL_t_speleotuning_1σ = rawD_SL[:,7]./1000; # from years to kyrs
```

In [47]:

```
# Plot of SprattLisiecki GSL

plot_SprattLisiecki_raw =
plot(# title = "SprattLisiecki GSL stack",
    xlabel = "Age (kyrs BP)",
    ylabel = "GSL (m NN)",
    size = (1000, 200))
plot!(t_SL_, GSL_mean_SL_,
    markersize = 0.1,
    ribbon = (GSL_mean_SL_ .- GSL_SL_err_lo_, GSL_SL_err_up_ .- GSL_mean_SL_), #
95% confidence interval SL from Monte Carlo analysis
    color = :darkblue,
    label = "Spratt & Lisiecki",
    legend = :topleft,
    bg_legend = :transparent)
```

Out[47]:



> SprattLisiecki is tuned to LR04 (meaning it has relicts of orbital tuning). Jo has tuned the stack to the Sanbao speleothem, to have the GSL stack on the same age model as the Mediterranean record (Rohling). SHOULD WE USE Jo's AGE MODEL?

ii) Reverse dataset, to redefine from age to time

In [59]:

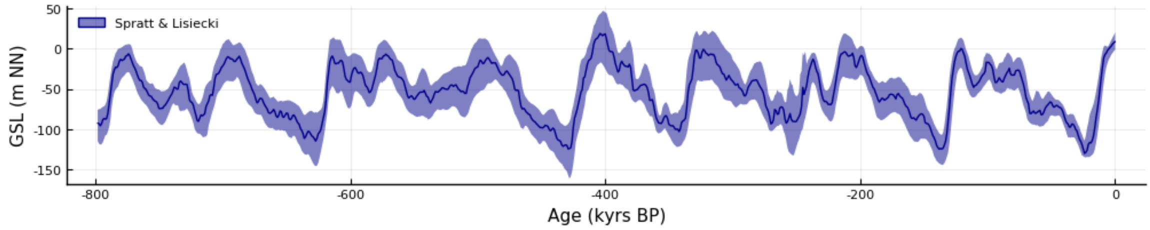
```
# For the analysis, we want to change age (increasing backwards) to time (increa
sing forward).
# We therefore reverse the dataset along the 1st dimension (rows)

revD_SL = reverse(rawD_SL, dims = 1) # reversing the dataset along the 1st dimen
sion (rows)

# Naming the arrays
t_SL = - revD_SL[:,1]/1000   # time (kyrs)    # (converted from years to kyrs by
dividing with 1000. Negative since we define present as 0)
GSL_mean_SL = revD_SL[:,2]   # SeaLev_longPC1 # indexation has been reversed. St
ays positive values
GSL_1σ_SL = revD_SL[:,3]     #                # indexation has been reversed. St
ays positive values   # USE THIS?
GSL_SL_err_lo = revD_SL[:,4] #                # from MonteCarlo analysis (lower
 95% CI quantile)      # OR THIS?
GSL_SL_err_up = revD_SL[:,5] #                # from MonteCarlo analysis (upper
 95% CI quantile)
;
# SprattLisiecki is tuned to LR04 (meaning it has relicts of orbital tuning)
# Jo has tuned the stack to the Sanbao speleothem, to have the GSL stack on the
 same age model as the Mediterranean record (Rohling)
## SHOULD WE USE Jo's AGE MODEL?
SL_t_speleotuning = -revD_SL[:,6]./1000 #from years to kyrs
SL_t_speleotuning_1σ = revD_SL[:,7]./1000; # from years to kyrs
```

In [81]:

```
@save "../Koding/WrangledDataFiles/BasicArrays/SprattLisiecki.jld2" t_SL GSL_mea
n_SL GSL_1σ_SL GSL_SL_err_lo GSL_SL_err_up SL_t_speleotuning SL_t_speleotuning_1
σ
```

In [74]:

```
# Plot of SprattLisiecki GSL

plot_SprattLisiecki_raw_time =
plot(# title = "SprattLisiecki GSL stack",
    xlabel = "Age (kyrs BP)",
    ylabel = "GSL (m NN)",
    size = (1000, 200))
plot!(t_SL, GSL_mean_SL,
    markersize = 0.1,
    ribbon = (GSL_mean_SL .- GSL_SL_err_lo, GSL_SL_err_up .- GSL_mean_SL), # 95%
confidence interval SL from Monte Carlo analysis
    color = :darkblue,
    label = "Spratt & Lisiecki",
    legend = :topleft,
    bg_legend = :transparent)

savefig("../../MASTER_2.0/figurar/RawData/GSL/SL_raw_time.pdf")
```



In [73]:

```
mean(GSL_1σ_SL) # (Article reads 1σ "Bootstrapping and random sampling yield mea
n uncertainty estimates of 9-12 m (1σ)").
# Checks out, kind of?
```

Out[73]:

```
12.070150187734665
```

bla # DISCUSSION on how to define the 95% confidence interval (2σ) # Plot plot(# title = "SprattLisiecki GSL stack", xlabel = "Time (kyrs BP)", ylabel = "GSL (m NN)", size = (1000, 200)) plot!(t_SL, GSL_mean_SL, markersize = 0.1, ribbon = (GSL_mean_SL .- GSL_SL_err_lo, GSL_SL_err_up .- GSL_mean_SL), # Looks more or less like the 2σ ribbon color = "blue", label = "Spratt&Lisiecki err_lo, err_up") plot!(t_SL, GSL_mean_SL, markersize = 0.1, ribbon = (GSL_1σ_SL, GSL_1σ_SL), label = "Spratt&Lisiecki 1σ") plot!(t_SL, GSL_mean_SL, markersize = 0.1, ribbon = (2 * GSL_1σ_SL), label = "Spratt&Lisiecki 2σ") # Note: looks like the quantiles given in dataset are for 95% CI. (matches 2σ) # How to most accurately describe 2σ? # GSL_2σ_SL_v1 = (GSL_SL_err_up .- GSL_SL_err_lo) / 2 # From MonteCarlo analysis # GSL_2σ_SL_v2 = 2 * GSL_1σ_SL # # are these the same ? # they look the same from the plot, but they are not exactly the same # GSL_2σ_SL_v1 .- GSL_2σ_SL_v2 # (If they were, this should give an array of zeros) # SO WHICH ONE CHOULD WE CARRY ON IN uivD? ### NONE OF THESE, uivD is defined with 1σ. 95% CI is then calculated after.

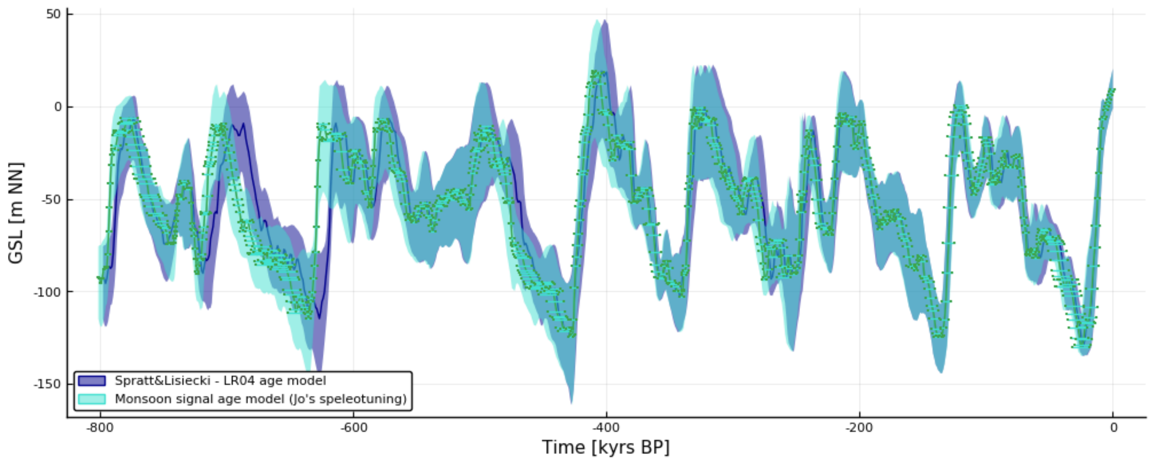
Notes on age model and uncertainties

The Spratt & Lisiecki record is tuned to the LR04 age model. As far as we can read, the LR04 age model uncertainties have already been incorporated in the dataset by Spratt & Lisiecki. We do not have a separate array for age uncertainties, as these are incorporated in the value-error through Monte Carlo analysis (equivalent to what we do with binned resampling).

> SprattLisiecki is tuned to LR04 (meaning it has relicts of orbital tuning)
>
> Jo has tuned the stack to the Sanbao speleothem, to have the GSL stack on the same age model as the Mediterranean and Red Sea RSL records (Rohling and Grant)
>
> SHOULD WE USE Jo's AGE MODEL?

In [97]:

```
# Plot of SprattLisiecki GSL
# Comparison of LR04 age model and Jo's tuning to monsoon signal in the Sanbao s
peleothem

plot_SprattLisiecki_agemodelcomparison =
plot(# title = "SprattLisiecki GSL stack",
    xlabel = "Time [kyrs BP]",
    ylabel = "GSL [m NN]", #NN given in unit in dataset -what does it mean?
    size = (1000, 400),
    legend = :bottomleft,
    #bg_legend = :transparent
    )
# Original LR04 age model
plot!(t_SL, GSL_mean_SL,
    markersize = 0.1,
    ribbon = (GSL_mean_SL .- GSL_SL_err_lo, GSL_SL_err_up .- GSL_mean_SL), # use
this or 2 * 1sigma?
    color = :darkblue,
    label = "Spratt&Lisiecki - LR04 age model"
)
# Speleotuning age model
plot!(SL_t_speleotuning, GSL_mean_SL, ribbon = (GSL_mean_SL .- GSL_SL_err_lo, GS
L_SL_err_up .- GSL_mean_SL), color = :turquoise, fillalpha = 0.5, label = "Monso
on signal age model (Jo's speleotuning)")
plot!(SL_t_speleotuning, GSL_mean_SL, xerr = 2*SL_t_speleotuning_1σ, ms = 1, msc
olor = :turquoise,  label = "")#"speleotuning age uncertainty (±2σ)")

savefig("../../MASTER_2.0/figurar/RawData/GSL/SL_agemodelcomparison.pdf")
```

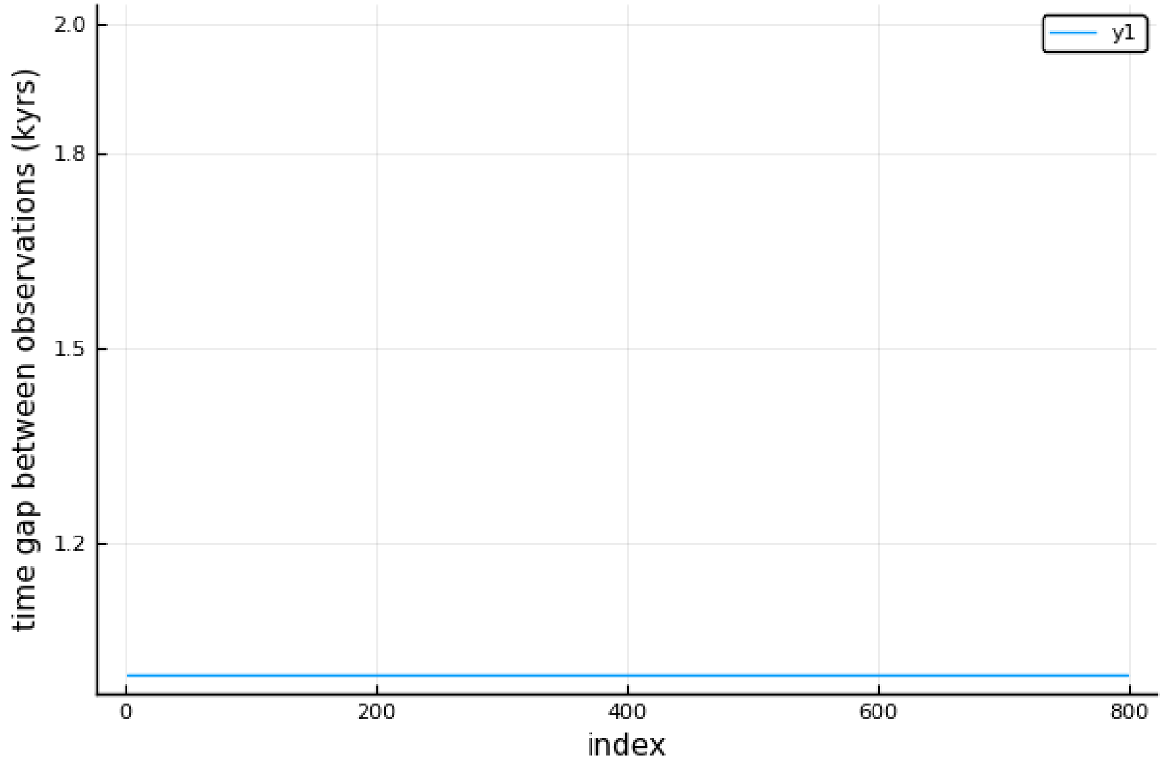


Interpolation?

In [76]:

```
# need for interpolation?
minimum(diff(t_SL))
mean(diff(t_SL))
maximum(diff(t_SL))

plot(diff(t_SL),
xlabel = "index",
ylabel = "time gap between observations (kyrs)")
```

Out[76]:



No need to interpolate the SprattLisiecki record, as it is already on a regular grid with one value every 1000 year-grid

v) Redefining `rD_SprattLisiecki` as an `UncertainIndexValueDataset`

In [ ]:

```
# Redefining ``rD_SprattLisiecki`` as an ``UncertainIndexValueDataset``

t_uiv_SL_noageunc = [UncertainValue(Normal, t_SL[i], 0) for i in 1:length(t_SL)]
# no age uncertainty
GSL_uiv_SL = [UncertainValue(Normal, GSL_mean_SL[i], GSL_1σ_SL[i]) for i in 1:le
ngth(GSL_mean_SL)]
uivD_SL_noageunc = UncertainIndexValueDataset(t_uiv_SL, GSL_uiv_SL)


# with age uncertainty from LR04
t_uiv_SL_ageunc = [UncertainValue(Normal, t_SL[i], t_1σ_SL[i]) for i in 1:length
(t_SL)] # SYSTEMATIC AGE UNCERTAINTY IN LR04  -INCLUDE HERE?
uivD_SL_ageunc = UncertainIndexValueDataset(t_uiv_SL_ageunc, GSL_uiv_SL)

# plot(uivD_SL)
```

In [ ]:

```
# Save the relevant arrays of the LR04 record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/SprattLisiecki.jld2" t_SL
uivD_SL_ageunc uivD_SL_noageunc
```

In [ ]:

```
plot(uivD_SL)
```

vii) Binned resampling on a 1 kyr timestep grid

In [ ]:

```
# Define grid for binned resampling
grid_SL = ceil(minimum(t_SL)) + binsize/2 : binsize : floor(maximum(t_SL)) - bin
size/2

# Define the resampling method to draw 1000 resamples within each bin of the gri
d, and return a median with associated confidence
resampling_method_SL = BinnedResampling(grid_SL, 1000) #  within each bin of the
grid_SL, draw a 1000 samples (with substitution)
```

Resample the uivD using the resampling method, to return one **mean/median** value with associated confidence for each bin in the grid

In [ ]:

```
@time SL_binned_fullength_ageunc = resample(uivD_SL_ageunc, resampling_method_SL
)
```

In [ ]:

```
@time SL_binned_fullength_noageunc = resample(uivD_SL_noageunc, resampling_metho
d_SL)
```

In [ ]:

```
@save "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/SprattLisie
cki.jld2" SL_binned_fullength_ageunc SL_binned_fullength_noageunc
```

In [ ]:

```
@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/SprattLisie
cki.jld2"
```

In [ ]:

```
plot(SL_binned_fullength_noageunc)
```

Above is a plot of the record with fixed time (uncertainty in the time dimension is transposed to uncertainty in the value dimension through binned resampling).

Next, we compute the 0.5 and 0.975 quantiles, and make a plot with the 95% confidence interval ribbon.

In [ ]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval

# Without the LR04 age model uncertainty
SL = SL_binned_fullength_noageunc


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

binmidpoints_SL = [SL.indices[i].value for i in 1:length(SL)]


plot_SL_binned_noageunc =
plot(binmidpoints_SL, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :darkblue,
    label = "Spratt & Lisiecki",
    xlabel = "Time [years BP]",
    ylabel = "GSL [m]",
    grid = false
    )
```

In [ ]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
SL = SL_binned_fullength_ageunc


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

binmidpoints_SL = [SL.indices[i].value for i in 1:length(SL)]


plot_SL_binned_ageunc =
plot(binmidpoints_SL, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :darkblue,
    label = "Spratt & Lisiecki",
    xlabel = "Time [years BP]",
    ylabel = "GSL [m]",
    grid = false
    )
```

2.2 - Elderfield GSL-record

- GSL record from *Elderfield et al. (2012)*, denoted **E** for Elderfield
- Mg/Ca temperature deconvolution of $\delta^{18}O_b$, spanning the last 1.5 Myr.
- Data available from Pangaea, DOI: https://doi.org/10.1594/PANGAEA.786204 (https://doi.org/10.1594/PANGAEA.786204)

In [109]:

```
# read in data
filepath_Elderfield = "../../MASTER_2.0/data/sea-level/d18O_ESL_Elderfield.tab"
rawD_E = readdlm(filepath_Elderfield, skipstart = 16)
revD_E = reverse(rawD_E, dims = 1)

# name columns
t_E = -revD_E[:,1]     # Age [ka BP]            # note: AGE [ka BP] is interpol
ated age. From LR04? CHECK
#revD[:,2]             # δ18O H2O [‰ SMOW]      # What is this?
GSL_E = -revD_E[:,3]  # Sea lev rel [m]        # NOTE: MADE NEGATIVE TO MATCH
 WITH OTHERS.
;
# uncertainty?? IMPORT AGE UNCERTAINTY FROM LR04?
# t_\sigma_LR04          # need to define anew. > <
```

Uncertainties

For the Elderfield record, no uncertainty is included in the public dataset. However, a note is made in the Elderfield article (2012), as well as an uncertainty analysis for this record is done in supplementary materials in Rohling et al. (2014). We will discuss these in the following.

Elderfield et al. (2012) report an "error in $\delta^{18}O_W$ of ± 0.2 $\textperthousand$ (per mille), from propagation of estimated temperature and $\delta^{18}O_C$ uncertainties.

- We interpret this as 1σ, and
- since $\delta^{18}O_W$ is the ice volume component of the signal, we define this same error on the GSL estimate.

In [106]:

```
# Create an array for 1σ based on the uncertainty communicated in Elderfield et
 al. (2012)
GSL_1σ_E___ = 0.0002 .* GSL_E # +- 0.2 per mille
 ;  # THIS LOOKS WAY TOO SMALL in plot, but is all I can gather from the articl
e. Something I'm not understanding?
```

Rohling et al. (2014) make a note of that uncertainties are poorly constrained in the Elderfield et al. (2012). They have therefore performed a probabilistic assessment of the Elderfield record. Rohling et al. (2014) find that the total uncertainty is of about ±35 m (1σ):

- The bulk of the uncertainty (± 35 m) is *random calibration uncertainty* on temperature sensitivity (T_S), due to many unknowns.
- In addition there is the $\delta^{18}O_W$ *to sea-level* conversion uncertainty (± 10% (0.1 meters uncertainty per meter GSL change is the standard ratio).

Rohling et al. (2012) further state that this uncertainty "may appear large, but there is strong autocorrelation in the record, which leads to considerably tighter uncertainty limits to underlying 'mean' trends."

Sinces it is not the aboslute sea-level we are interested in, but rather the *dynamics* (relative changes) in sea level, we therefore don't include the random calibration uncertainty, but only the sea-level conversion uncertainty.

In [110]:

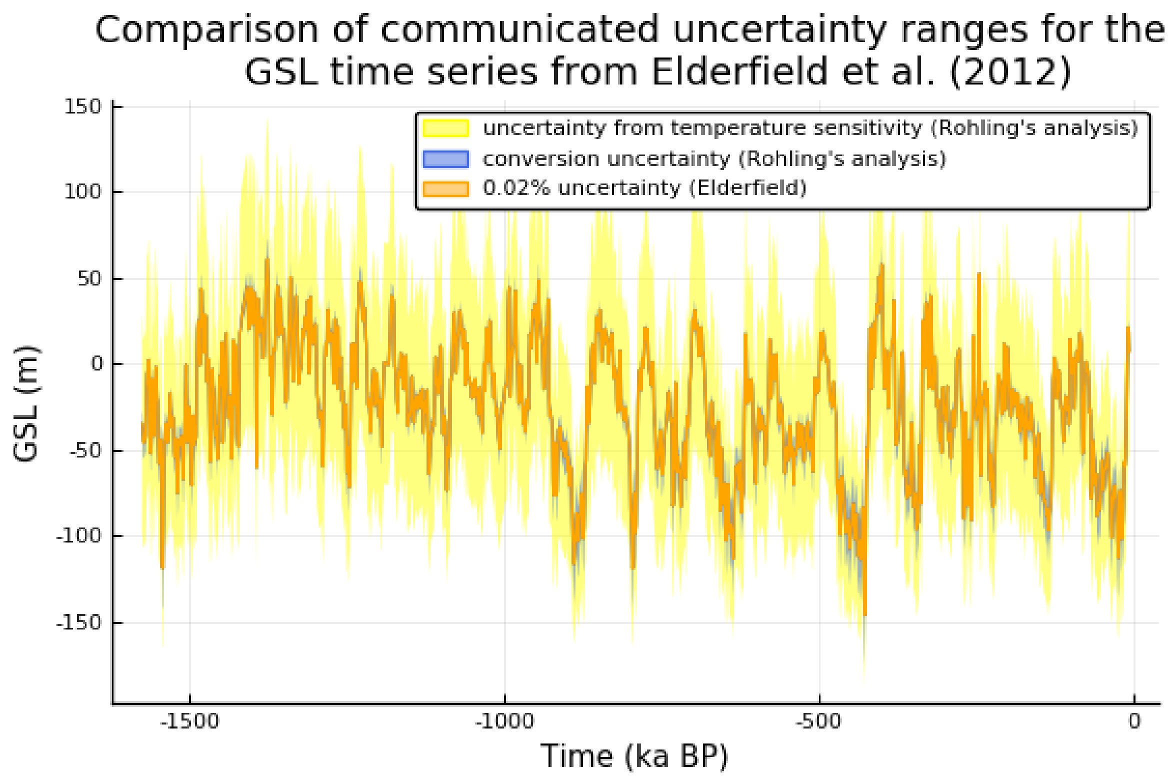
```
# Create an array for 1σ based on the uncertainty communicated in Rohling et al
 (2014)

GSL_1σ_E = zeros(length(GSL_E)) # Create an empty array of the length of the rec
ord, to contain the 1σ


unc_calibrationTs = GSL_1σ_E[:] .= 35  # Uncertainty from temperature sensitivit
y calibration: adds up to ±35 meters GSL  # BUT WE DON'T WANT TO USE THIS
unc_conversion = GSL_E .* 0.1           # Uncertainty from conversion of d18O_W t
o sea-level:   1σ = 10%



GSL_1σ_E_hiunc = unc_calibrationTs .+ unc_conversion # hiunc = high uncertainty
 - including uncertainties on temperature sensistivity
GSL_1σ_E_lounc = unc_conversion                      # lounc = low uncertainty
 - excluding uncertainties on temperature sensistivity
;
```

> **Note**: weird that Elderfield communicates a 2%% (per mille) conversion uncertainty, and Rohling 10% conversion uncertainty. Comments?

In [109]:

```
# plot for visual comparison of the communicated uncertainty ranges

plot(title = "Comparison of communicated uncertainty ranges for the
    GSL time series from Elderfield et al. (2012)",
    xlabel = "Time (ka BP)",
    ylabel = "GSL (m)")
plot!(t_E, GSL_E,
    ribbon = (2*GSL_1σ_E_hiunc), # 95% confidence interval. Using the general 1σ
= +-35 meters, this adds up to +-140 meters...
    color = :yellow,
    label = "uncertainty from temperature sensitivity (Rohling's analysis)")
plot!(t_E, GSL_E,
    ribbon = (2*GSL_1σ_E_lounc), # 95% CI. Using only conversion uncertainty (1
\sigma is 10%)
    color = :royalblue,
    label = "conversion uncertainty (Rohling's analysis)")
plot!(t_E, GSL_E,
    ribbon = (2*GSL_1σ_E___), # 95% CI. 1\sigma is 0.02% (0.02 permille)
    color = :orange,
    label = "0.02% uncertainty (Elderfield)")
```

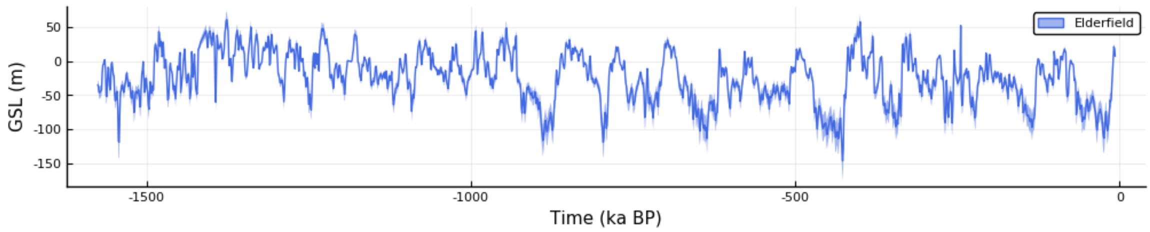
Out[109]:



Note: We only take the *conversion uncertainty* (lounc, ±10%) into account in running our analysis. We take care not to translate the uncertainty in determining the temperature sensitivity into uncertainty in the dynamics of sea level change (this would be an overcommunication of uncertainty). The reason for this is that we are not really interested in the absolute sea level change, but rather the *dynamics* of sea-level change.

In [111]:

```
# plot the Elderfield time series with the value-uncertainties we will use

plot_Elderfield_raw_noageunc =
plot(#title = "GSL time series from Elderfield et al. (2012)",
    xlabel = "Time (ka BP)",
    ylabel = "GSL (m)",
    size = (1000,200))
plot!(t_E, GSL_E,
    ribbon = (2*GSL_1σ_E_lounc), #plotting the 95% confidence interval. Using on
ly conversion uncertainty
    color = :royalblue,
    label = "Elderfield")

savefig("../../MASTER_2.0/figurar/RawData/GSL/plot_Elderfield_noageunc.pdf")
```



Note on age model uncertainties

The Elderfield GSL record is correlated to the LR04 stack. Age model uncertainties on the LR04 stack are read in above. However, we make a note of the fact that the Elderfield record is a continuous stratigraphic core, implying next to no uncertainty in relative chronology. Therefore, propagations of age uncertainties from the LR04 age model should be done strictly monotonic (no age reversals allowed).

> What will be the right way to include age uncertainties? Include in uivD or not? How does the BinnedResampling method treat age uncertainties?
>
> Veiledningsmøte 9/6/2020: **Include age uncertainties in uivD** for now. We can later run sensitivity analyses to check if we get very different results with/without age uncertainties.

An array for the age model uncertainties will be created below, with the interpolation

In [121]:

```
# Make an array of the potential systematic deviations in the LR04 age model

t_4σ_LR04 = zeros(length(t_LR04)) # maximum age model envelope (interpreted as t
he 95% confidence envelope, i.e, ±2σ)

t_4σ_LR04[ t_LR04 .> -1000] .= 4                              # 4 ky from -1 Ma to 0
 (present)
t_4σ_LR04[(t_LR04 .<= -1000) .& (t_LR04 .> -3000)] .= 6  # 6 ky from -3 to -1 Ma
t_4σ_LR04[(t_LR04 .<= -3000) .& (t_LR04 .> -4000)] .= 15 # 6 ky from -4 to -3 Ma
t_4σ_LR04[(t_LR04 .<= -4000) .& (t_LR04 .> -5000)] .= 30 # 6 ky from -3 to -1 Ma
t_4σ_LR04[ t_LR04 .<= -5000] .= 40                            # 40 ky for before -5 M
a

t_1σ_LR04 = t_4σ_LR04 ./ 4 # 1σ
;

# interpolation

# create a continuous function with linear interpolation between every datapoint
in the array.
# This allows us to choose the interpolated value for any point in time.
interpolate_t_1σ_LR04 = LinearInterpolation(t_1σ_LR04, t_LR04)     # age model u
ncertainty (full)

# make a fine grained grid to contain the interpolated values
#fine_grid_LR04 = ceil(minimum(t_LR04)) : 0.1 : floor(maximum(t_LR04)) # One bin
for every 100 years (0.1 kyrs)

# make new arrays for interpolated data
# give a value from the interpolate function to every bin in the fine grid
#intpD_t_1σ_LR04 = [interpolate_t_1σ_LR04(i) for i in fine_grid_LR04]

# age model uncertainty from LR04
t_1σ_E = [interpolate_t_1σ_LR04(i) for i in t_E]


############## OR,  define  age model uncertainty directly
t_4σ_E = zeros(length(t_E)) # maximum age model envelope (interpreted as the 95%
confidence envelope, i.e, ±2σ)
t_4σ_E[ t_E .> -1000] .= 4                          # 4 ky from -1 Ma to 0 (present)
t_4σ_E[(t_E .<= -1000) .& (t_E .> -3000)] .= 6  # 6 ky from -3 to -1 Ma

t_1σ_E = t_4σ_E ./ 4 # 1σ
```

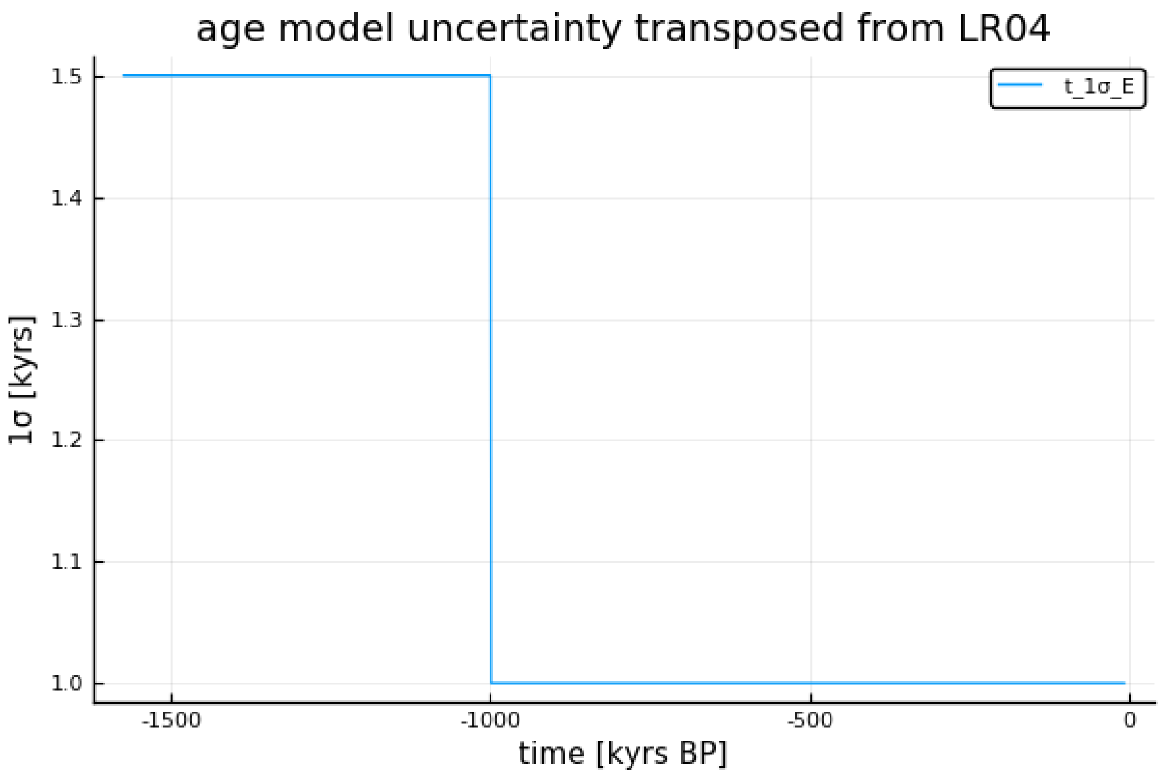
Out[121]:

```
1557-element Array{Float64,1}:
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 1.5
 ⋮
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
 1.0
```

In [122]:

```
plot(t_E, t_1σ_E, xlabel = "time [kyrs BP]", ylabel = "1σ [kyrs]", label = " t_1σ_E", title =  "age model uncertainty transposed from LR04" )
```

Out[122]:



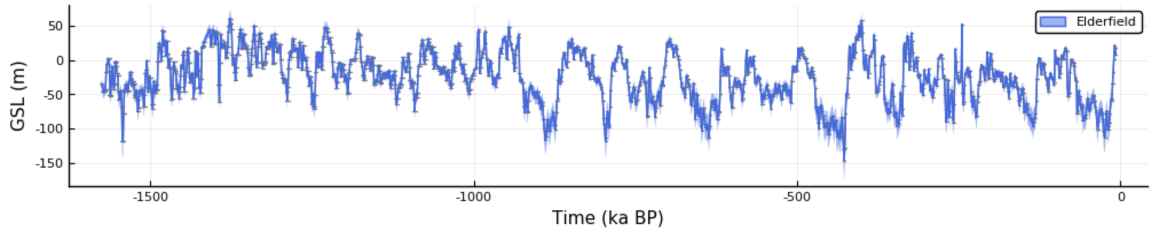
We see that we get 1σ = 1 kyr for the youngest 1 Ma part of the record, and 1σ = 1.5 kyr for the remaining part of the record. That is right according to the 95% confidence interval (4σ) being 4 kyrs for 0-1 Ma and 6 kyrs for 1-3 Ma. We are thus sure to have created the age model uncertainty array correctly.

In [115]:

```
@save "../Koding/WrangledDataFiles/BasicArrays/Elderfield.jld2" t_E t_1σ_E GSL_E
GSL_1σ_E_lounc
```

In [114]:

```
# plot the Elderfield time series with the LR04 age model uncertainty
plot_Elderfield_raw_ageunc =
plot(#title = "GSL time series from Elderfield et al. (2012)",
    xlabel = "Time [ka BP]",
    ylabel = "GSL [m]",
    size = (1000,200))
plot!(t_E, GSL_E, xerr = (2*t_1σ_E, 2*t_1σ_E), # ± 2σ
    ms=0.1, color= :grey,  label = "")#"LR04 age model uncertainty")
plot!(t_E, GSL_E,
    ribbon = (2*GSL_1σ_E_lounc), #plotting the 95% confidence interval. Using on
ly conversion uncertainty
    color = :royalblue,
    label = "Elderfield")

savefig("../../MASTER_2.0/figurar/RawData/GSL/plot_Elderfield_ageunc.pdf")
```



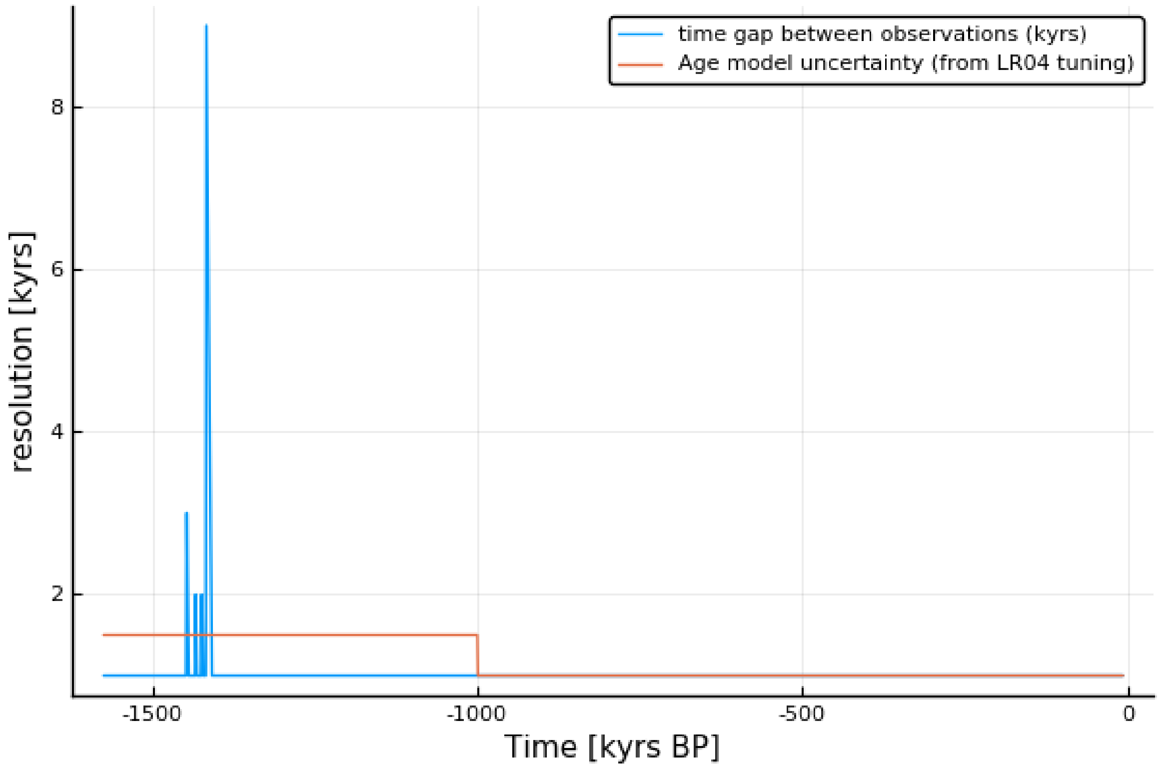
iv) Interpolation

First, check if there is need for interpolation. (We want at least a millenial resolution for our analysis)

In [123]:

```
# need for interpolation?

minimum(diff(t_E)) # 1000 years
mean(diff(t_E))     # 1007 years
maximum(diff(t_E)) # 9000 years
# Yes, we need to interpolate

plot(t_E, diff(t_E),
xlabel = "Time [kyrs BP]",
ylabel = "resolution [kyrs]",
label = "time gap between observations (kyrs)",
)
plot!(t_E, t_1σ_E, label = "Age model uncertainty (from LR04 tuning)")
```

Out[123]:



Mean time gap between observations is around 1000 years. The largest time gap between observations is 9000 years. To meet the requirement of the common grid we will later define we therefore need to interpolate, to fill in the missing values.

In the case of this record, the highest resolution is one value every 1000 years. I will therefore not do higher resolution analyses with this record.

> LOGICAL QUESTION: making a higher resolution interpolation grid will be misleading when it comes to the resampling, won't it? Since it will estimate uncertainties based on fake data. Won't it be better here to use the interpolation grid as 1 kyr resolution?

In [ ]:

```
# Interpolation of Elderfield record

interpolate_t_E  = LinearInterpolation(t_E, t_E)       # interpolation function
 for time
interpolate_E    = LinearInterpolation(GSL_E, t_E)     # interpolation function
 for GSL

fine_grid_E = minimum(t_E) : 1 : maximum(t_E)        # fine grained grid for whi
ch we want to interpolate values - one bin for every 100 years
                                                     # MAYBE WE SHOULD JUST HAVE
1 kyr interpolation? HIGHER WOULD ONLY BE FAKE DATA...

intpD_t_E = [interpolate_t_E(i) for i in fine_grid_E]  # Array of the interpolat
ed time values
intpD_GSL_E = [interpolate_E(i) for i in fine_grid_E]; # Array of the interpolat
ed  GSL values



# carrying on equivalent array of uncertainties
intpD_1σ_E = abs.(intpD_GSL_E) .* 0.1 # conversion uncertainty +/- 10% (lounc) #
WHAT's THE abs FOR??

# age model uncertainty from LR04
intpD_t_1σ_E = [interpolate_t_1σ_LR04(i) for i in fine_grid_E]
;# like discussed above, this looks right
```

In [ ]:

```
intpD_t_E
```

In [ ]:

```
# plot interpolated array on top of recorded values, to check that it correspond
s
plot(title = "Elderfield GSL - comparison of original and interpolated time seri
es",
    xlabel = "Time (ka BP)",
    ylabel = "GSL (m)",
    size = (1000,200))
scatter!(t_E, GSL_E,
    ribbon = (GSL_1σ_E_lounc),
    ms = 1,
    label = "Elderfield observations")
plot!(intpD_t_E, intpD_GSL_E,
    ribbon = (intpD_1σ_E),
    label = "Elderfield interpolated")
# ok
```

In [ ]:

```
# plot interpolated array on top of observational values, to check that it corre
sponds

plot_Elderfield_raw_unc =
plot(#title = "Elderfield GSL with uncertainties",
    xlabel = "Time (ka BP)",
    ylabel = "GSL (m)",
    size = (1000,200))
plot!(intpD_t_E, intpD_GSL_E,
    xerr = intpD_t_1σ_E,
    label = "age uncertainty from LR04")
scatter!(t_E, GSL_E,
    ribbon = (intpD_1σ_E),
    label = "Elderfield et al. (2012)",
    color = :royalblue)
```

v) Redefine the interpolated array as an `UncertainIndexValueDataset`

In [ ]:

```
# Redefining Elderfield interpolated data as uivD

# Note, here lounc is used (only d18O conversion uncertainty, not temperature se
nsitivity),
GSL_uiv_E = [UncertainValue(Normal, intpD_GSL_E[i], intpD_1σ_E[i]) for i in 1:le
ngth(intpD_GSL_E)]  # conversion uncertainty 10%, as reported in analysis of Roh
ling et al. (2014)

# 2 time arrays are defined:
# - one with no uncertainty in time dimension
t_uiv_E_noageunc = [UncertainValue(Normal, intpD_t_E[i], 0) for i in 1:length(in
tpD_t_E)] # LR04 age model uncertainty not included

# - and one with age model uncertainty from LR04
t_uiv_E_ageunc = [UncertainValue(Normal, intpD_t_E[i], intpD_t_1σ_E[i]) for i in
1:length(intpD_t_E)]  # time uncertainty from LR04 age model

uivD_E_ageunc = UncertainIndexValueDataset(t_uiv_E_ageunc, GSL_uiv_E)
uivD_E_noageunc = UncertainIndexValueDataset(t_uiv_E_noageunc, GSL_uiv_E)
```

In [ ]:

```
@time plot(uivD_E)
```

Save the relevant arrays of the wrangled Elderfield data in a .jld2 file

In [ ]:

```
# Save the relevant arrays of the Elderfield record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Elderfield.jld2" intpD_t_
E uivD_E_ageunc uivD_E_noageunc
```

vi) Binned resampling on a grid timesteps of 1000 years

In [ ]:

```
#Define the resampling method for the Elderfield record

# define grid
tmin_E = ceil(minimum(intpD_t_E)) + binsize/2
tmax_E = floor(maximum(intpD_t_E)) - binsize/2
grid_E = tmin_E : binsize : tmax_E

# In each bin of the grid, resample 1000 values
resampling_method_E = BinnedResampling(grid_E, 1000)
```

In [ ]:

```
# resample the uncertain index value dataset with the resampling method defined
 above
@time E_binned_fullength_ageunc = resample(uivD_E_ageunc, resampling_method_E)
```

In [ ]:

```
# resample the uncertain index value dataset with the resampling method defined
 above
@time E_binned_fullength_noageunc = resample(uivD_E_noageunc, resampling_method_
E)
```

In [ ]:

```
# Save the relevant arrays of the Elderfield record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Elderfield.
jld2" E_binned_fullength_ageunc E_binned_fullength_noageunc
```

In [ ]:

```
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/Elderfield.jld2"
```

Plot of 95% confidence interval, including the LR04 age model uncertainty:

In [ ]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
E = E_binned_fullength_ageunc

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

# time array
binmidpoints_E = [E.indices[i].value for i in 1:length(E)]


plot_E_binned_ageunc =
plot(binmidpoints_E, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :royalblue,
    label = "Elderfield",
    xlabel = "Time [years BP]",
    ylabel = "GSL [m]",
    grid = false
    )
```

Without the LR04 age model uncertainty:

In [ ]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
E = E_binned_fullength_noageunc

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

# time array
binmidpoints_E = [E.indices[i].value for i in 1:length(E)]


plot_E_binned_noageunc =
plot(binmidpoints_E, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :royalblue,
    label = "Elderfield",
    xlabel = "Time [years BP]",
    ylabel = "GSL [m]",
    grid = false
    )
```

The difference seems small, even though we initially estimated the LR04 age model uncertainty to be overcommunicated.

We will use the version with age model uncertainty in our first round of analysis, for a conservative approach.

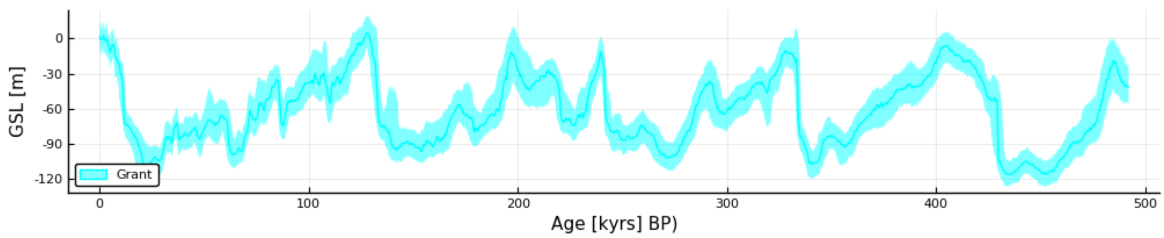
2.3 Grant RSL/GSL-record

- *Grant et al. (2014)*, denoted **G**
- High resolution record of relative sea level at the straight of Bab el Mandab, Red Sea (equivalent to GSL). Spans the last 500 kyr.
- Data available from Nature, DOI: https://doi.org/10.1038/ncomms6076 (https://doi.org/10.1038/ncomms6076)

In [123]:

```
# read in data
filepath_Grant = "../../MASTER_2.0/data/sea-level/Grant(2014)/SD1_v1.txt"
rawD_G = readdlm(filepath_Grant, dims = (3951-14, 4), '\t', '\r', skipstart = 14
)

# naming columns
t_G_ = rawD_G[:,1]        # time_kyr
RSL_G_ = rawD_G[:,4]       # RSL_Pmax_m
RSL_q95lo_G_ = rawD_G[:,2] # RSL_95low_m
RSL_q95up_G_ = rawD_G[:,3] # RSL_95high_m
# define an array to carry on uncertainty - standard deviation (1σ):
RSL_1σ_G_ = (RSL_q95up_G_ .- RSL_q95lo_G_) / 4 # 95%CI-quantiles include 2σ on e
ach side of mean, so we divide uncertainty range by 4 to get 1σ (standard deviat
ion)
;

# plot
plot_Grant_raw_age =
plot(#title = "Grant RSL record",
    xlabel = "Age [kyrs] BP)",
    ylabel = "GSL [m]",
    size = (1000,200),
    legend = :bottomleft,
    #bg_legend = :transparent
)
plot!(t_G_, RSL_G_,
    ribbon = (RSL_G_ .- RSL_q95lo_G_, RSL_q95up_G_ .- RSL_G_), #plotting the 95%
confidence interval
    color = :cyan,
    label = "Grant")
```

Out[123]:



Reverse dataset to redefine from age to time

In [132]:

```
revD_G = reverse(rawD_G, dims = 1)
typeof(revD_G)  # Array{Float64,2}

# naming columns
t_G = -revD_G[:,1]         # time_kyr
RSL_G = revD_G[:,4]        # RSL_Pmax_m
RSL_q95lo_G = revD_G[:,2] # RSL_95low_m
RSL_q95up_G = revD_G[:,3] # RSL_95high_m
# define an array to carry on uncertainty - standard deviation (1σ):
RSL_1σ_G = (RSL_q95up_G .- RSL_q95lo_G) ./ 4 # 95%CI-quantiles include 2σ on eac
h side of mean, so we divide uncertainty range by 4 to get 1σ (standard deviatio
n)
;
```

In [145]:

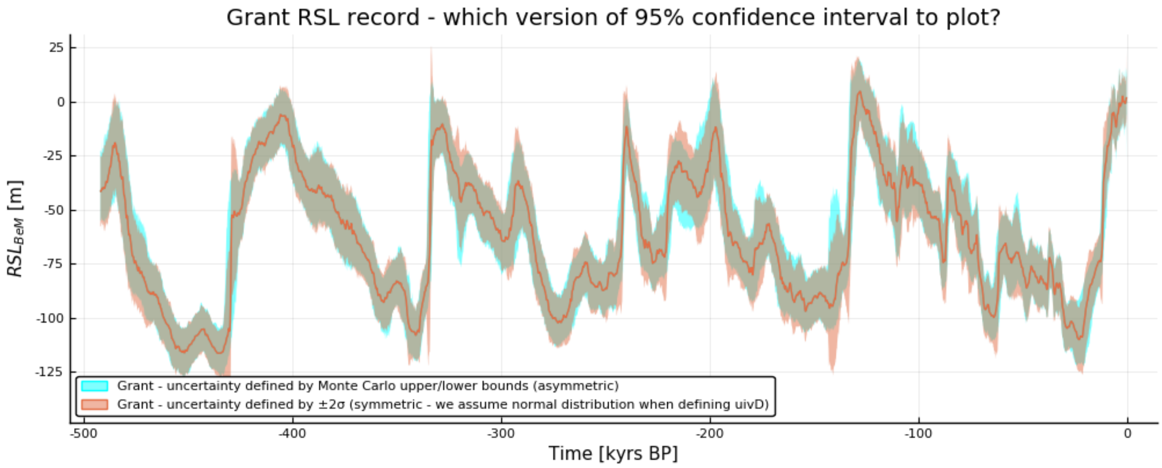
```
@save "../Koding/WrangledDataFiles/BasicArrays/Grant.jld2" t_G RSL_G RSL_1σ_G RS
L_q95lo_G RSL_q95up_G
```

Age uncertainty in the Grant record is incorporated in the value dimension through Monte Carlo analysis (like we do when binned resampling)

In [143]:

```
# plot
plot_Grant_raw =
plot(title = "Grant RSL record - which version of 95% confidence interval to plo
t?",
    xlabel = "Time [kyrs BP]",
    ylabel = string(L"RSL_{BeM}", " [m]"),
    size = (1000,400),
    legend = :bottomleft,

    )
plot!(t_G, RSL_G,
    ribbon = (RSL_G - RSL_q95lo_G, RSL_q95up_G - RSL_G), #plotting the 95% confi
dence interval
    color = :cyan,
    label = "Grant - uncertainty defined by Monte Carlo upper/lower bounds (asym
metric)")

plot!(t_G, RSL_G, ribbon = (2*RSL_1σ_G, 2*RSL_1σ_G), label = "Grant - uncertaint
y defined by ±2σ (symmetric - we assume normal distribution when defining uivD)"
)
```

Out[143]:



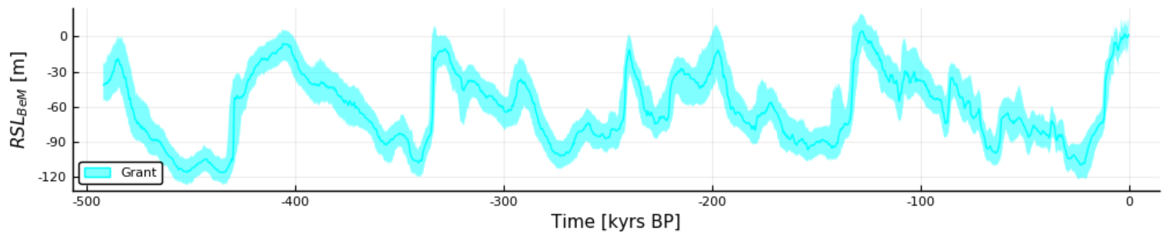
Which version of the 95% confidence interval should we plot?

- Monte carlo analysis (reported in dataset give slightly asymmetric uncertainty,
- while we assume normal distribution (σ)

In [140]:

```
#@load "../Koding/WrangledDataFiles/BasicArrays/Grant.jld2" t_G RSL_G RSL_1σ_G

# plot
plot_Grant_raw =
plot(xlabel = "Time [kyrs BP]",
    ylabel = string(L"RSL_{BeM}", " [m]"),
    size = (1000,200),
    legend = :bottomleft,
    )
plot!(t_G, RSL_G,
    ribbon = (RSL_G - RSL_q95lo_G, RSL_q95up_G - RSL_G), #plotting the 95% confi
dence interval
    color = :cyan,
    label = "Grant")

#savefig("../../MASTER_2.0/figurar/RawData/GSL/plot_Grant.pdf")
```

Out[140]:



Interpolation. First, check if there is need for interpolation on the Grant data. (We want at least a millenial resolution for our analysis)

In [ ]:

```
# need for interpolation?

minimum(diff(t_G)) # 125 years
mean(diff(t_G))    # 125 years
maximum(diff(t_G)) # 125 years

plot(diff(t_G),
xlabel = "index",
ylabel = "time gap between observations (kyrs)",
label = "diff(t_G)")
```

Grant is already on a regular time grid, with one observation every 125 years. We therefore have no need for interpolation on this dataset. We also note that we may use this time series to run higher resolution analyses.

Redefine as an `UncertainIndexValueDataset`

In [ ]:

```
t_uiv_G = [UncertainValue(Normal, t_G[i], 0) for i in 1:length(t_G)] # no age un
certainty - incorporated in RSL uncertainty through monte carlo analysis
RSL_uiv_G = [UncertainValue(Normal, RSL_G[i], RSL_1σ_G[i]) for i in 1:length(RSL
_G)]
uivD_G = UncertainIndexValueDataset(t_uiv_G, RSL_uiv_G)
```

In [ ]:

```
# @time plot(uivD_G, xlabel = "Time [kyrs]", ylabel = "GSL [m]", label = "Gran
t", title = "Grant uivD
    (uncertainties in x and y directions carried in kernel density estimates)")
```

In [ ]:

```
# Save the relevant arrays of the Elderfield record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/WrangledData_Grant.jld2" t_G ui
vD_G
```

vi) Binned resampling

First, we bin the record on a grid with timesteps of 1000 years

In [15]:

```
@load "../../MASTER_2.0/Koding/WrangledDataFiles/WrangledData_Grant.jld2"
```

Out[15]:

```
2-element Array{Symbol,1}:
 :t_G
 :uivD_G
```

In [ ]:

```
tmin_G = ceil(minimum(t_G))
tmax_G = floor(maximum(t_G))
grid_G = tmin_G + binsize/2 : binsize : tmax_G - binsize/2

resampling_method_G = BinnedResampling(grid_G, 1000)
```

In [ ]:

```
G_binned_fullength = resample(uivD_G, resampling_method_G)
```

Plots

Plot of the uivD, here we see time uncertainty has been transposed into value uncertainty through the binned resampling

In [ ]:

```
# plot(uivD_G)
a = rand(10)
plot(a, grid = (1,0))
```

> I want some help to plot with a 1 kyr grid, to illustrate what happens when binned resampling

In [ ]:

```
#plot(G_binned_fullength, grid = ())
```

Now we plot the binned resampled uivD time series with the 95% confidence interval

In [ ]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
G = G_binned_fullength

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

# time array
binmidpoints_G = [G.indices[i].value for i in 1:length(G)]


plot_G_binned =
plot(binmidpoints_G, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :cyan,
    label = "Grant",
    xlabel = "Time [years BP]",
    ylabel = string(L"RSL_{BeM}", " [m]"),
    grid = false
    )
```

Binned resampling of the record on a finer grid, to use in high resolution analyses. We will make two hr versions; one with timestep of 125 years, for analyses with La2004, and one with timesteps of 500 years, for high resolution analysis with the Martinez-García record.

- `G_binned_fullength_hr125` , for hr analyses with La2004

In [ ]:

```
# Binned resampling on a finer grid, for high resolution analyses

binsize_hr = 0.125
grid_G_hr = minimum(t_G) + binsize_hr/2 : binsize_hr : maximum(t_G) - binsize_hr
/2
# Intuitively, we would have the grid start at half a binsize *before* the first
datapoint (binmidpoint).
# But that might skew the first point in the record a bit (unless we extrapolate
for the first half of the bin, but that also imposes some assumptions).
# We therefore opt to cut the length of the timeseries by a datapoint in each en
d, to avoid skewedness or extrapolation.

resampling_method_G_hr = BinnedResampling(grid_G_hr, 1000)

G_binned_fullength_hr125 = resample(uivD_G, resampling_method_G_hr)
```

In [ ]:

```
### Plot the high resolution binned resampled uivD with it's 95% confidence inte
rval
G = G_binned_fullength_hr

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

# time array
binmidpoints_G = [G.indices[i].value for i in 1:length(G)]


plot_G_binned =
plot(binmidpoints_G, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :cyan,
    label = "Grant",
    xlabel = "Time [years BP]",
    ylabel = string(L"RSL_{BeM}", " [m]"),
    grid = false
    )
```

- `G_binned_fullength_hr500` , for high resolution analysis with the Martinez-García record.

In [18]:

```
# Binned resampling on finer grid, for high resolution analyses with Martinez-Ga
rcía

binsize_hr = 0.5 # time step 0.5 kyr
grid_G_hr = minimum(t_G) + binsize_hr/2 : binsize_hr : maximum(t_G) - binsize_hr
/2
# Intuitively, we would have the grid start at half a binsize *before* the first
datapoint (binmidpoint).
# But that might skew the first point in the record a bit (unless we extrapolate
for the first half of the bin, but that also imposes some assumptions).
# We therefore opt to cut the length of the timeseries by a datapoint in each en
d, to avoid skewedness or extrapolation.

resampling_method_G_hr500 = BinnedResampling(grid_G_hr, 1000)

G_binned_fullength_hr500 = resample(uivD_G, resampling_method_G_hr500)
```

Out[18]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 983 uncertain values coupled with 983 uncertain indic
es
```

In [19]:

```
# Save the relevant arrays of the Elderfield record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Grant.jld2"
G_binned_fullength G_binned_fullength_hr125 G_binned_fullength_hr500
```

In [20]:

```
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/Grant.jld2" # This will b
e read in  in our Toolbox notebook (NB3)
```

Out[20]:

```
3-element Array{Symbol,1}:
 :G_binned_fullength
 :G_binned_fullength_hr125
 :G_binned_fullength_hr500
```

1.4 Rohling (2014)

- denoted *R*
- Relative sea-level at straight of Gibraltar, record spanning 5.3 Ma.
- Data available from... DOI..
- Note main text caveats and reasoning for which parts of the records will be used: Sapropelic intervals makes the record inapt for the the post-MPT. We could opt to exclude the sapropelic intervals, but this would call for interpolation over large time gaps, and compromise the confidence level of the results. Therefore, we leave the record as is, and interpret the results with a pinch of salt.

i) Read in data

In [159]:

```
# read in data
filepath_Rohling = "../../MASTER_2.0/data/sea-level/Rohling(2014)/Rohling(2014)_
data_fig2_v1_rslGib.txt"
# .txt file with column 1-6 from the published dataset.
rawD_Rohling = readdlm(filepath_Rohling,'\t', Any, '\r', skipstart =  8, dims =
(5331, 6))

# naming columns
# First two columns are Wang et al. Discontinuous due to sapropel intervals. We
 will instead use the probabilistic analysis, column 3-6.
t_R_sapgap = rawD_Rohling[:,1]    # A: Med.RSL chronology (ky)
RSL_R_sapgap = rawD_Rohling[:,2]    # B: Med RSL (m; gaps are sapropel intervals)

# ...we use the Probabilistic analysis of the Wang record, by Rohling et al. (20
14)
t_R_ =  rawD_Rohling[:,3]        # chronology (ky)
RSL_mean_R_ = rawD_Rohling[:,4] # Med. RSL MEDIAN        # (m; shifted so mean o
f last three ky =0) (??)
q95_up_R_ = rawD_Rohling[:,5]    # upper 95%CI quantile    # Med. RSL upper bound
 95% probability interval for the median (~equivalent to 2se)
q95_lo_R_ = rawD_Rohling[:,6]    # lower 95%CI quantile    # Med. RSL lower bound
 95% probability interval for the median (~equivalent to 2se)
;

# plot Rohling
plot_Rohling_raw_age =
plot(#title = "Mediterranean sea level stack (Rohling)",
    size = (1000,200),
    xlabel = "Age (kyrs)",
    ylabel = "Sea level (m)")
plot!(t_R_, RSL_mean_R_,
    markersize = 0.5,
    ribbon = (RSL_mean_R_ .- q95_lo_R_, q95_up_R_ .- RSL_mean_R_), # 95% CI
    color = :skyblue,
    label = "Rohling")
plot!(t_R_sapgap, RSL_R_sapgap)
```

```
Cannot convert SubString{String} to series data for plotting

Stacktrace:
 [1] error(::String) at ./error.jl:33
 [2] prepareSeriesData(::SubString{String}) at /Users/maria/.julia/p
ackages/Plots/qZHsp/src/series.jl:14
 [3] convertToAnyVector(::SubString{String}, ::Dict{Symbol,Any}) at
/Users/maria/.julia/packages/Plots/qZHsp/src/series.jl:26
 [4] (::getfield(Plots, Symbol("##152#155")){Dict{Symbol,Any}})(::Su
bString{String}) at ./none:0
 [5] iterate(::Base.Generator{Array{Any,1},getfield(Plots, Symbol("#
#152#155")){Dict{Symbol,Any}}}, ::Int64) at ./generator.jl:47
 [6] append_any(::Any, ::Vararg{Any,N} where N) at ./essentials.jl:7
28
 [7] convertToAnyVector(::Array{Any,1}, ::Dict{Symbol,Any}) at /User
s/maria/.julia/packages/Plots/qZHsp/src/series.jl:41
 [8] macro expansion at /Users/maria/.julia/packages/Plots/qZHsp/sr
c/series.jl:129 [inlined]
 [9] apply_recipe(::Dict{Symbol,Any}, ::Type{Plots.SliceIt}, ::Array
{Any,1}, ::Array{Any,1}, ::Nothing) at /Users/maria/.julia/packages/
RecipesBase/zBoFG/src/RecipesBase.jl:275
 [10] _process_userrecipes(::Plots.Plot{Plots.PyPlotBackend}, ::Dict
{Symbol,Any}, ::Tuple{Array{Any,1},Array{Any,1}}) at /Users/maria/.j
ulia/packages/Plots/qZHsp/src/pipeline.jl:83
 [11] _plot!(::Plots.Plot{Plots.PyPlotBackend}, ::Dict{Symbol,Any},
::Tuple{Array{Any,1},Array{Any,1}}) at /Users/maria/.julia/packages/
Plots/qZHsp/src/plot.jl:178
 [12] #plot!#143(::Base.Iterators.Pairs{Union{},Union{},Tuple{},Name
dTuple{(),Tuple{}}}, ::typeof(plot!), ::Plots.Plot{Plots.PyPlotBacke
nd}, ::Array{Any,1}, ::Vararg{Array{Any,1},N} where N) at /Users/mar
ia/.julia/packages/Plots/qZHsp/src/plot.jl:158
 [13] plot!(::Plots.Plot{Plots.PyPlotBackend}, ::Array{Any,1}, ::Var
arg{Array{Any,1},N} where N) at /Users/maria/.julia/packages/Plots/q
ZHsp/src/plot.jl:155
 [14] #plot!#142(::Base.Iterators.Pairs{Union{},Union{},Tuple{},Name
dTuple{(),Tuple{}}}, ::typeof(plot!), ::Array{Any,1}, ::Vararg{Array
{Any,1},N} where N) at /Users/maria/.julia/packages/Plots/qZHsp/src/
plot.jl:150
 [15] plot!(::Array{Any,1}, ::Array{Any,1}) at /Users/maria/.julia/p
ackages/Plots/qZHsp/src/plot.jl:144
 [16] top-level scope at In[159]:29
```

ii) Redefine from age to time by reversing the dataset

In [152]:

```
# Reverse the dataset
revD_Rohling = reverse(rawD_Rohling, dims = 1 )

#t_R_sapgap = -revD_Rohling[:,1]    # A: Med.RSL chronology (ky)
#RSL_R_sapgap = revD_Rohling[:,2]   # B: Med RSL (m; gaps are sapropel interval
s)

t_R = -revD_Rohling[:,3]       # chronology (ky)        # negative, so that time
runs forward with increasing indices
RSL_mean_R = revD_Rohling[:,4] # Med. RSL MEDIAN        # (m; shifted so mean of
last three ky =0) (??)
q95_up_R = revD_Rohling[:,5]   # upper 95%CI quantile   # Med. RSL upper bound 9
5% probability interval for the median (~equivalent to 2se)
q95_lo_R = revD_Rohling[:,6]   # lower 95%CI quantile   # Med. RSL lower bound 9
5% probability interval for the median (~equivalent to 2se)

# Reformulate the uncertainties from quantiles to standard deviance (1σ)
RSL_1σ_R = (q95_up_R .- q95_lo_R) / 4
;
```

Age uncertainty has been incorporated in the RSL standard deviation through Monte Carlo probabilistic analysis.

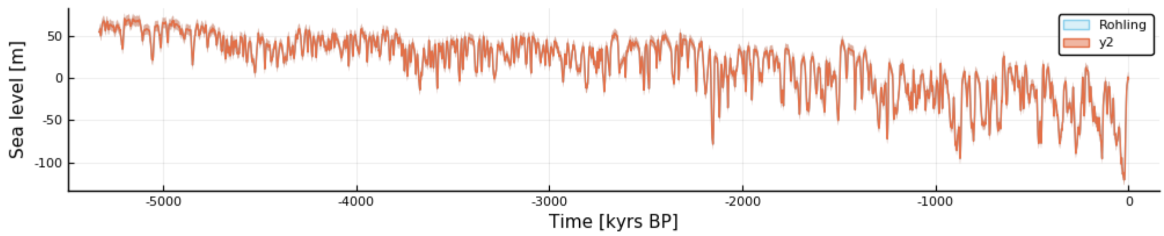
In [162]:

```
@save "../Koding/WrangledDataFiles/BasicArrays/Rohling.jld2" t_R RSL_mean_R q95_
lo_R q95_up_R
```

In [163]:

```
# plot Rohling
plot_Rohling_raw =
plot(#title = "Mediterranean sea level stack (Rohling)",
    size = (1000,200),
    xlabel = "Time [kyrs BP]",
    ylabel = "Sea level [m]")
plot!(t_R, RSL_mean_R,
    ribbon = (RSL_mean_R .- q95_lo_R, q95_up_R .- RSL_mean_R),
    fillalpha = 0.3,
    color = :skyblue,
    label = "Rohling")
#plot!(t_R, RSL_mean_R, ribbon = (2*RSL_1σ_R)   ) # Checking that same as above,
 95% CI. All good.

savefig("../../MASTER_2.0/figurar/RawData/GSL/plot_Rohling_raw.pdf")
```

Out[163]:



Interpolation

Note on sapropelic intervals

The Rohling record has a significant challenge, that it is riddled with **sapropelic intervals**. These intervals are anomalies in the GSL record, hypothesized to be periods of major surface freshwater dilution, and so the $\delta^{18}O_p$ signal is not indicative of sea level for these periods. For the Grant record, there were records keeping track of sapropelic intervals was kept for each individual datapoint ("using colour, core-scanning X-ray fluorescence(XRF), and magnetic data, in addition to stable isotope, organic carbon, and microfossil abundance data from previous studies"), allowing for a precise identification and exclusion of sapropelic intervals from the record. This does not exist for the Rohling record, however, and the *sapropelic intervals of the Mediterranean has instead been identified and removed through a signal processing approach.*

They also report that 3 sapropel intervals seem to not have been filtered out by the signal processing. These were detected by discord with other GSL records overlapping in time. A caveat to analyses with this record is that we presently have no way to know if there are more sapropelic intervals that has not been captured by the signal processing. Since this is the only high resolution GSL record we have that spans further back than Elderfield (1.5 Ma), such intervals would not be identified further back in the record.

> would like to at least mark the sapropelic intervals, to showcase the problem. Have tried, but need help on how to.

In [160]:

```
t_R_sapgap;
```

In [161]:

```
using Missings
df = collect(Missings.replace(t_R_sapgap, NaN))
;
```

HELP. # MISSING VALUES in the first two columns (Sapropel intervals). Need to be dealt with #import Pkg; Pkg.add("Missings") using Missings Missings.replace(x, 1) Missings.EachReplaceMissing{Array{Union{Missing, Int64},1},Int64}(Union{Missing, Int64}[1, 2, missing], 1) julia> collect(Missings.replace(x, 1)) 3-element Array{Int64,1}: 1 2 1 julia> collect(Missings.replace(x, 1)) == coalesce.(x, 1) true

Missing values in the first two columns of the dataset (renamed `sapgap`) are sapropel intervals.

In [ ]:

```
plot(t_R_sapgap, diff(t_R_sapgap)) # Doesn't work because haven't replaced missi
ng values
```

Explore the resolution of the Rohling record when sapropelic intervals are cut out

In [ ]:

```
### Explore the resolution of the Rohling record when sapropelic intervals are c
ut out ###

# Read in data from Rohling et al.
#### Note: I have cut out the sapropel intervals (marked by gaps in the first co
lumn) from this file.####
sapcutfile = "../../MASTER_2.0/data/sea-level/Rohling(2014)/Rohling(2014)_data_f
ig2_sapropelscutout.txt"
rawD_Rohling_sapcut = readdlm(sapcutfile,'\t', Any, '\r', skipstart =  8, dims =
(3736, 6))

# Reversing the dataset
revD_Rohling_sapcut = reverse(rawD_Rohling_sapcut, dims = 1 )# to convert from a
ge (increasing backwards) to time (increasing forward, with present defined as
 0)

# naming columns of interest
t_R_sapcut = -revD_Rohling_sapcut[:,3]       # chronology (ky)
RSL_mean_R_sapcut = revD_Rohling_sapcut[:,4] # Med. RSL MEDIAN         # (m; shif
ted so mean of last three ky =0) (??)
q95_up_R_sapcut = revD_Rohling_sapcut[:,5]   # upper 95%CI quantile    # Med. RSL
upper bound 95% probability interval for the median (~equivalent to 2se)
q95_lo_R_sapcut = revD_Rohling_sapcut[:,6]   # lower 95%CI quantile    # Med. RSL
lower bound 95% probability interval for the median (~equivalent to 2se)
# Reformulate the uncertainties from quantiles to standard deviance (1σ)
RSL_1σ_R_sapcut = (q95_up_R_sapcut .- q95_lo_R_sapcut) / 4
;

# Explore the resolution of the Rohling record when sapropelic intervals are cut
out

print("The highest resolution after sapropel intervals cut out is ", minimum(dif
f(t_R_sapcut)), "
","Average resolution after sapropel intervals are cut out is ", mean(diff(t_R_s
apcut)), "
","Sapropelic intervals create gaps in the record of up to ", maximum(diff(t_R_s
apcut)), " kyrs.")

plot(t_R_sapcut, diff(t_R_sapcut),
    xlabel = "Age [kyrs BP]",
    ylabel = "time gap [kyrs]",
    title = "Resolution of Rohling sea-level record
    after cutting out reported sapropelic intervals ")
```

The resolution plot above shows the intervals that were filtered out by signal processing (Rohling et al. (2014).

As mentioned, *Rohling et al.(2014)* also report that 3 sapropel intervals seem to not have been filtered out by the signal processing. I did not find the data necessary to mark these intervals, but they are roughly at 450, 550 and 750 kyrs BP (marked in the figures of the Rohling article, and it is also evident in the comparison with the other GSL plots, which we will see below). These anomalies make the Rohling record inapt for analyses across the post-MPT.

Interpolation?

We could opt to exclude these "residual" sapropelic intervals. However, this would call for interpolation over large time gaps, since our method requires continuous time series, which would compromise the confidence level of the results/statistical power of the method. (How much interpolation is acceptable before a breakdown of statistical power in the method could be explored further with a sensitivity analysis, but that is outside the scope of this project.) Given that we do not have access to the exact time intervals of the three later identified sapropelic intervals, we instead leave the record *as is*, with the sapropelic intervals, and omit using it for the post-MPT. Additionally, since we cannot be sure there are other sapropelic intervals not captured by the signal processing, we interpret any results from analyses with the Rohling record with a pinch of salt. This is thus one of the records where we have to *bite i det sure eplet* and accept the limitations of our data.

> Per now, there is one analysis with the Rohling record that gives significant results in the *opposite direction* from significant results with the other GSL records (Rohling → Bereiter on the postMPT_500 grid). **What to make of this, when both are *significant* results?**

Sapropelic intervals are reportedly identified mainly for the first 700 years of the record, and we therefore planned not to include the Rohling record for analyses on the post-MPT.

Interpolation

The dataset was originally given on a 1 kyr regular time grid. If we cut out the sapropel intervals, there are gaps in the record we need to interpolate to run the analysis.

In [ ]:

```
# We have decided not to cut out the 3 sapropelic intervals
# Evaluate need for interpolation by checking the resolution

minimum(diff(t_R)) #1
mean(diff(t_R))    #1
maximum(diff(t_R)) #1

plot(diff(t_R),
xlabel = "index",
ylabel = "time gap between observations (kyrs)",
label = "diff(t_R)")
```

The Rohling record is already on a regular 1 kyr grid, so no need for interpolation.

Redefining `revD_Rohling` as an `UncertainIndexValueDataset`

In [ ]:

```
# Refining data as an ``UncertainIndexValueDataset``

t_uiv_R = [UncertainValue(Normal, t_R[i], 0) for i in 1:length(t_R)]
RSL_uiv_R = [UncertainValue(Normal, RSL_mean_R[i], RSL_1σ_R[i]) for i in 1:lengt
h(RSL_mean_R)] #HELP ON DEFINING intpD_RSL_mean_R
uivD_R = UncertainIndexValueDataset(t_uiv_R, RSL_uiv_R)
;
```

In [ ]:

```
plot(uivD_R)
```

In [ ]:

```
# Save the relevant arrays of the Rohling record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Rohling.jld2" t_R RSL_mea
n_R RSL_1 uivD_R
```

vi) Binned resampling

In [ ]:

```
grid_R = t_R[1] - binsize/2 : binsize : t_R[end] + binsize/2
#= dette er ein ryddigare og finare måte å definere på
men då må eg kjøre alle binned resamplings på nytt
og ikkje noko problem med sånn det er gjort allereie
forutan at eg mister eit datapunkt i kvar ende
så kanskje berre Grant og Chalk det har noko for seg å redefinere =#
```

In [ ]:

```
grid_R = ceil(minimum(t_R)) + binsize/2 : binsize : floor(maximum(t_R)) - binsiz
e/2
```

In [ ]:

```
resampling_method_R = BinnedResampling(grid_R, 1000) # 1000 draws in each bin.

R_binned_fullength = resample(uivD_R, resampling_method_R)

@save "../Koding/WrangledDataFiles/Binned_ts_fullength/Rohling.jld2" R_binned_fu
llength
```

In [ ]:

```
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/Rohling.jld2"
```

In [ ]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
R = R_binned_full

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

# time array
binmidpoints_R = [R.indices[i].value for i in 1:length(R)]


plot_R_binned =
plot(binmidpoints_R, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :skyblue,
    label = "Rohling",
    xlabel = "Time [years BP]",
    ylabel = string(L"RSL_{Gib}", " [m]"),
    size = (1000,200),
    grid = false
    )
```

Comparative plots between GSL records

Comparative plot between SprattLisiecki and Rohling

In [ ]:

```
# Comparative plot between SprattLisiecki and Rohling

plot(#title = "Comparison of Spratt&Lisiecki's & Rohling's GSL records",
    xlabel = "Time (kyrs BP)",
    ylabel = "Sea level (m)",
    size = (1000,200),
    xtics = (0:100:1000),
    xlim = (-1000,0)) # Plot only last 1 Myr BP

plot!(t_R, RSL_mean_R,
    markersize = 0.5,
    ribbon = (RSL_mean_R .- q95_lo_R, q95_up_R .- RSL_mean_R),            # 95%
CI
    color = :skyblue,
    label = "Rohling")

plot!(t_SL, GSL_mean_SL,
    markersize = 0.5,
    ribbon = (GSL_mean_SL - GSL_SL_err_lo, GSL_SL_err_up - GSL_mean_SL),   # 95%
CI
    # xerr = t_σ_SL, # age uncertainty from Jo (speleothem tuning of record) - U
SE OR NOT?
    color = :darkblue,
    label = "Spratt & Lisiecki")
```

Large deviations in GSL dynamics between the records around 750, 550 and 450 kyrs BP. These seem to be concordant in timing with the potential bands of sapropelic intervals mentioned in Rohling et al. (2014) as not having been filtered out by the signal-processing (marked in yellow in the article, but I couldn't find the indices in the dataset.)

In any case, this illustrates the point that the GSL records available are *significantly different*. That is the reason why we will run our analyses with many different records for GLS, so that we can make more robust conclusions.

> Why are the uncertainties (confidence interval) so very different?

In [ ]:

```
# Plot to compare all GSL reconstructions

plot_GSL_comparative_1500kyrs =
plot(#title = "Comparative plot of GSL records",
    xlabel = "Time [ka BP]",
    ylabel = "GSL [m]",
    grid = true,         # not working
    xlim = (-1500, 0),   # plot only a range 0:1000 ka
    ticks = (0:100:-1000),# not working
    size = (1000,200),
    share = :(x,y),      # working?
    #legend = :bottomleft
    )
plot!(t_R, RSL_mean_R,
    ribbon = (2*RSL_1σ_R), # ±2σ, aka 95% confidence interval
    color = :skyblue,
    label = "Rohling"
    )
#plot!(fine_grid_E, intpD_GSL_E, yerr = (2*t_1σ_E), color = :royalblue, ms = 0.
1) # LR04 age model uncertainty
plot!(fine_grid_E, intpD_GSL_E,
    ribbon = (2*intpD_1σ_E),
    color = :royalblue,
    label = "Elderfield"
    )
#plot!(t_SL, GSL_mean_SL, yerr = (2*t_1σ_SL), color = :darkblue, ms = 0.1) # LR0
4 age model uncertainty
plot!(t_SL, GSL_mean_SL,
    ribbon = (GSL_mean_SL - GSL_SL_err_lo, GSL_SL_err_up - GSL_mean_SL), # 95% C
I
    color = :darkblue,
    label = "Spratt & Lisiecki"
    )
plot!(t_G, RSL_G,
    ribbon = (RSL_G - RSL_q95lo_G, RSL_q95up_G - RSL_G),                 # 95% C
I
    color = :cyan,
    label = "Grant"
    )

savefig("../figurar/RawData/GSL/plot_GSL_comparative_1500kyrs.pdf")
```

In [ ]:

```
# Compare GSL records with d18O record

plot(plot_GSL_comparative_full,
    plot_LR04_raw_time,
    layout = grid(2,1),
    link = :x)
```

In [ ]:

```
# Subplot of GSL records

plot_GSL_overview =
plot(plot_SprattLisiecki_raw,
    plot_Elderfield_raw_unc,
    plot_Grant_raw,
    plot_Rohling_raw,
    layout = (4,1),
    link = :x,
    #link = :(x, y),        # HOW TO link both x and y axis?
    size = (1000,400))
#plot!(title = "Global sea level records")

#savefig("../../Master_2.0/figurar/RawData/GSL/plot_GSL_overview.pdf")
```

> help on making it pretty (common xlabel, and maybe ylabel)

3 - Insolation

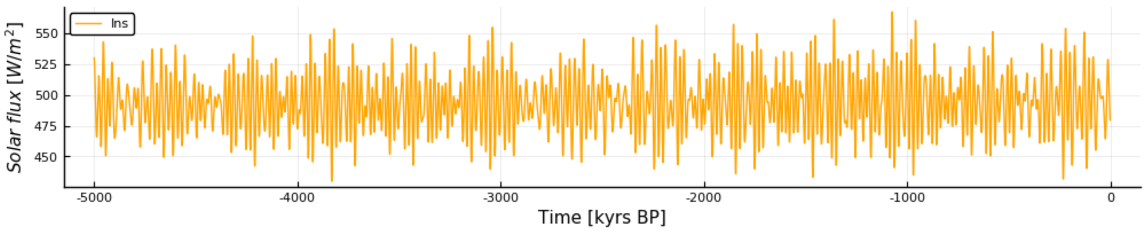
- Insolation record La2004, from *Laskar et al. (2004)* (denoted **Ins**).
- Numerical solution for midsummer insolation at 65˚N
- Time series computed using the AnalySeries computer software (Paillard et al., 1994)

In [43]:

```
# Read in La2004

filepath_insol = "../../MASTER_2.0/data/insolation/Paillard_computation/Ins_65N_Daily_ins21June_65N_past5Ma" # (nB) - La2004 solution, computed using AnalySeries
rawD_insol = readdlm(filepath_insol, dims = (50001,2), skipstart = 2)


# reverse columns, to redefine from age (ka BP, increasing backwards) to time (kyrs BP, increasing forwards)
revD_insol = reverse(rawD_insol, dims = 1)

# name columns of interest
t_insol   = -revD_insol[:,1] # time [kyrs BP]. Made negative to define 0 as present
insol_65N = revD_insol[:,2]  # insolation: daily mean TOA solar flux at 65N summer solstice, [W/m^2]

@save "../Koding/WrangledDataFiles/BasicArrays/La2004.jld2" t_insol insol_65N
```

In [44]:

```
# plot
plot_insolation =
plot(# title = "Insolation",
    xlabel = "Time [kyrs BP]",
    ylabel = L"Solar \ flux \ [W/m^2]",
    size = (1000,200))
plot!(t_insol, insol_65N,
    color = "orange",
    label = "Ins")

savefig("../../Master_2.0/figurar/RawData/Insolation/plot_La2004_65N.pdf")
```



iv) Interpolation to time grids equivalent to time steps we will use

Assigning the time series to grids with time steps we will use in analyses (1 kyr, 500 yrs, 125 yrs).

In [45]:

```
# interpolation on grids with time steps we will use

# Make a continuous function of interpolated values
interpolate_t_insol = LinearInterpolation(t_insol, t_insol)
interpolate_insol_65N = LinearInterpolation(insol_65N, t_insol)

grid_insol_1 = t_insol[1] : 1 : t_insol[end] # 1 kyr, common time step for all a
nalyses
# Arrays with interpolated values for timesteps of 500 years
La2004_t_fullength       = [interpolate_t_insol(i) for i in grid_insol_1]
# time array, interpolated to  high resolution
La2004_insol65N_fullength = [interpolate_insol_65N(i) for i in grid_insol_1]  #
 insolation values array, interpolated for 500 year timestep analyses

binsize_hr500 = 0.5 # for analyses with Lambert and Martinez-Garcia
grid_insol_hr500 = t_insol[1] : binsize_hr500 : t_insol[end] # we set the grid f
or the interpolated data to give values concordant with the Grant and Chalk reco
rds for the high resolution analyses
# Arrays with interpolated values for timesteps of 500 years
La2004_t_fullength_hr500        = [interpolate_t_insol(i) for i in grid_insol_hr
500]         # time array, interpolated to  high resolution
La2004_insol65N_fullength_hr500 = [interpolate_insol_65N(i) for i in grid_insol_
hr500]  # insolation values array, interpolated for 500 year timestep analyses

# make a grid onto which we assign values from the interpolated function
binsize_hr = 0.125 # for analyses with Grant and Chalk
grid_insol_cr = t_insol[1] : binsize_hr : t_insol[end]
# Arrays with interpolated values for timesteps of 125 years
La2004_t_fullength_hr125        = [interpolate_t_insol(i) for i in grid_insol_cr
]         # time array, interpolated to  high resolution
La2004_insol65N_fullength_hr125 = [interpolate_insol_65N(i) for i in grid_insol_
cr]  # insolation values array, interpolated for 125  year timestep analyses
;
```

v) carry on arrays to analysis Note: Computed insolation values has no associated uncertainty on this timescale (certain up to >50 Ma into past and present). We therefore have no need in redefining the insolation data as UncertainIndexValueDataset. We can instead carry on the arrays `t_insol` and `insol_65` directly to analysis. For high resolution analysis we carry on the equivalent interpolated arrays.

In [47]:

```
# Save the relevant arrays of the La2004 record in a .jld2 file

#### These are the arrays we will carry on for insolation:
# timestep 1 kyr - standard grid for analysis with all records
La2004_t_fullength             # age array to define common time interval
La2004_insol65N_fullength      # ... insolation values to be sent to causality tes
t
# interpolated arrays for high resolution analysis
# timestep 500 years - for hr analyses with Lambert, Martinez-Garcia and Bereite
r
La2004_t_fullength_hr500
La2004_insol65N_fullength_hr500
# timestep 125 years - for hr analysis with Chalk and Grant records
La2004_t_fullength_hr125
La2004_insol65N_fullength_hr125

@save "../../MASTER_2.0/Koding/WrangledDataFiles/La2004.jld2" La2004_t_fullength
La2004_insol65N_fullength  La2004_t_fullength_hr125 La2004_insol65N_fullength_hr
125 La2004_t_fullength_hr500 La2004_insol65N_fullength_hr500
```

In [48]:

```
# checking that all 6 arrays included in the file to be carried on in NB3
@load "../../MASTER_2.0/Koding/WrangledDataFiles/La2004.jld2"
# checking, all good
```

Out[48]:

```
6-element Array{Symbol,1}:
 :La2004_t_fullength
 :La2004_insol65N_fullength
 :La2004_t_fullength_hr125
 :La2004_insol65N_fullength_hr125
 :La2004_t_fullength_hr500
 :La2004_insol65N_fullength_hr500
```

4 - pCO2 records

These are the reference records we will use for pCO2:

- Bereiter et al (2015), spanning the last 800 kyrs
- Chalk et al (2017), spanning the interval from 1.090 - 1.240 Ma
- Hönisch et al (2009), spanning the last 2 Myrs.

4.1 Bereiter pCO2 record

- pCO2 record from *Bereiter et al. (2015)*, denoted **B**
- Direct measurement of pCO2 concetration in air bubbles from Epica Dome C (EDC) ice core.
- denoted B
- age model AICC2012
- Data available from... **- got the file from Jo, where to reference?**

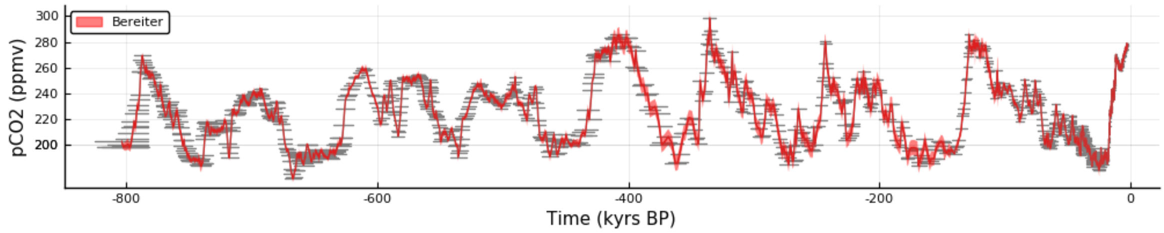
In [3]:

```
# Read in data from Bereiter
# AICC2012 age model
filepath_Bereiter = "../../MASTER_2.0/data/CO2/Bereiter2015_CO2_ageunc.txt"
rawD_B = readdlm(filepath_Bereiter, '\t', Float64,'\n', dims = (1643,4), skipsta
rt = 1)# reading in the file
revD_B = reverse(rawD_B, dims = 1) # reversing along the first dimension (rows)

# naming columns
t_B = -(revD_B[:,1] /1000) # 1st column = Time (kyrs) - divided by 1000 to chang
e unit from years to kyrs. Negative because we define present as 0 kyrs.
t_1σ_B = revD_B[:,2] /1000 # 2nd column = age model uncertainty - divided by 100
0 to change unit from years to kyrs. σ remains a positive value
CO2_mean_B = revD_B[:,3]   # 3rd column = pC02 (ppmv) - remains in positive valu
es
CO2_1σ_B = revD_B[:,4]     # 4th column = pCO2 uncertainty (ppmv) -  σ remains a
positive value
;

@save "../Koding/WrangledDataFiles/BasicArrays/Bereiter.jld2" t_B t_1σ_B CO2_mea
n_B CO2_1σ_B
```

In [4]:

```
# overview plot of pCO2
@load "../Koding/WrangledDataFiles/BasicArrays/Bereiter.jld2" t_B t_1σ_B CO2_mea
n_B CO2_1σ_B

#plot_Bereiter_raw_ageunc =
plot(#title = "800 kyrs of pCO2",
    size = (1000, 200),
    xlabel = "Time (kyrs BP)",
    ylabel = "pCO2 (ppmv)",
    legend = :topleft)
plot!(#title = "Bereiter pCO2 record, from Epica Dome C ice core"
    t_B, CO2_mean_B, xerr = (2 * t_1σ_B), # use 1 or 2 σ for?
    ms = 0.1, color = :grey,
    label = "", #"Age model uncertainty (±2σ)
    )
plot!(t_B, CO2_mean_B,
    ribbon = (2 * CO2_1σ_B),  # 95% CI
    color = "red",
    alpha = 0.5,
    label = "Bereiter"
    )
```

Out[4]:



In [40]:

```
savefig("../../Master_2.0/figurar/RawData/pCO2/Bereiter_raw_timeunc.pdf")
```

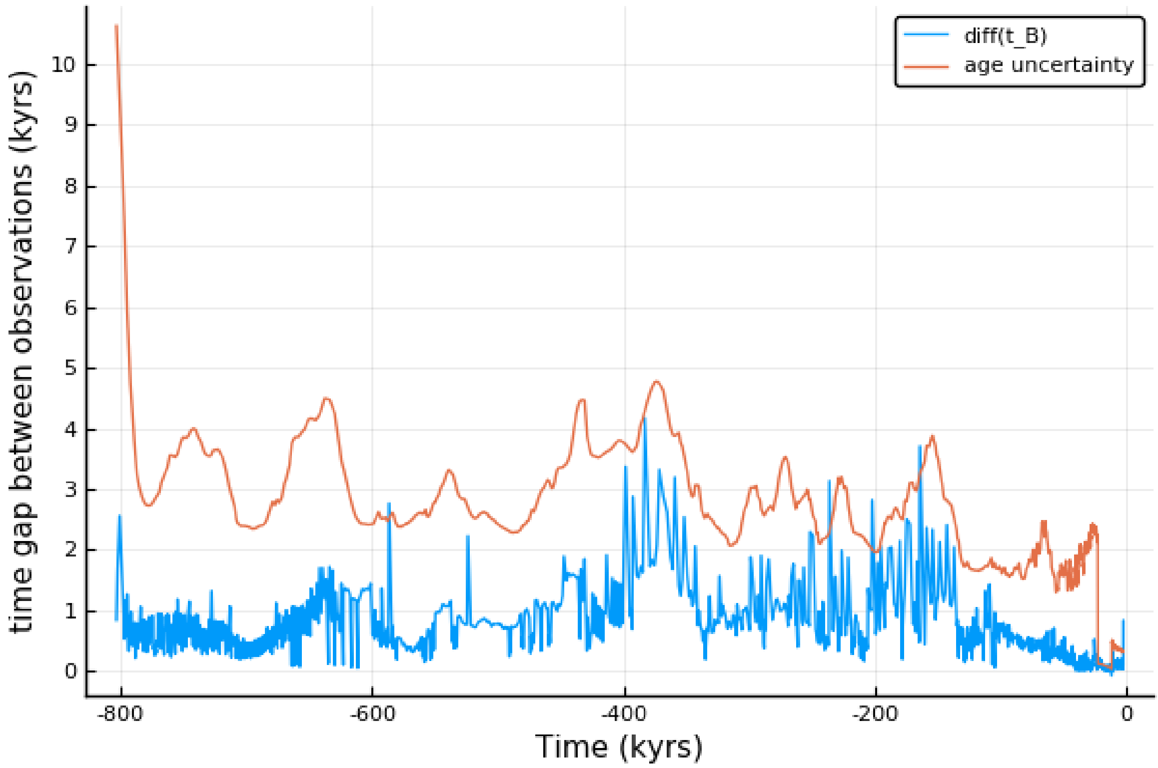
iv) Interpolation

First, we check if there is need for interpolation of this record (We have decided a 1 kyr resolution for our analyses)

In [5]:

```
# Need for interpolation?
minimum(diff(t_B))  # 0.07
mean(diff(t_B))     # 490 years
maximum(diff(t_B))  # >4000 years
# so yes we need to interpolate, because our grid for analysis calls for a value
every 1000 years.


plot(t_B, diff(t_B),
    xlabel = "Time (kyrs)",
    ylabel = "time gap between observations (kyrs)",
    label = "diff(t_B)",
    ytick = (0:1:10))
plot!(t_B,t_1σ_B, label = "age uncertainty")
```

Out[5]:



The Bereiter record has irregular resolution, with gaps of up to 4000 years between observations. However, the age uncertainty is large enought to get some values in each bin through the binned resampling.

With a mean uncertainty of 490 years, we may also try out higher resolution analyses. We will however set the mean resolution as a lower boundary of the time steps used in analyses.

Redefining as `UncertainIndexValueDataset`

We redefine the arrays to an uivD, which carries the uncertainties as kernel density estimates (KDE). The binned resampling function can then resample from the probability distribution.

1. uivD including age model uncertainty

In [6]:

```
# Redefining as uivD_B

t_uiv_B = [UncertainValue(Normal, t_B[i], t_1σ_B[i]) for i in 1:length(t_B)]
CO2_uiv_B = [UncertainValue(Normal, CO2_mean_B[i], CO2_1σ_B[i]) for i in 1:lengt
h(CO2_mean_B)]
uivD_B = UncertainIndexValueDataset(t_uiv_B, CO2_uiv_B)
```

Out[6]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1643 uncertain values coupled with 1643 uncertain ind
ices
```

Note: The Bereiter (pCO2) and Lambert (dust) records are both from the same ice core site (EDC). For analyses between these two time series, we can therefore exclude uncertainties in the age model. We therefore also prepare a version of the uivD that doesn't include uncertainties in age.

1. uivD without age model uncertainty

In [7]:

```
# uivD_B_EDC defined without age uncertainty, for analysis with other records fr
om the EDC core.

t_uiv_B_EDC = [UncertainValue(Normal, t_B[i], 0) for i in 1:length(t_B)] # age m
odel uncertainty not included
CO2_uiv_B = [UncertainValue(Normal, CO2_mean_B[i], CO2_1σ_B[i]) for i in 1:lengt
h(CO2_mean_B)]
uivD_B_EDC = UncertainIndexValueDataset(t_uiv_B_EDC, CO2_uiv_B)

# Save the relevant arrays of the Bereiter record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Bereiter_nointp.jld2" uiv
D_B uivD_B_EDC
```

In [ ]:

```
# plot(uivD_B)
```

vi) Binned resampling on grid

In [8]:

```
@load "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Bereiter_nointp.jld2"
```

Out[8]:

```
2-element Array{Symbol,1}:
 :uivD_B
 :uivD_B_EDC
```

In [9]:

```
uivD_B
#= Be sure to use the original data array
(original array (without interpolation) has 1643 datapoints
interpolated array has 8018 datapoints
=#
```

Out[9]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1643 uncertain values coupled with 1643 uncertain ind
ices
```

1. Binned resampling with age model uncertainty

In [13]:

```
# binned resamplig on a 1 kyr grid

binsize = 1 # 1 kyr
tmin_B = ceil(minimum(t_B))   # first bin midpoint at a whole kyr
tmax_B = floor(maximum(t_B))  # last bon midpoint at a whole kyr
grid_B = tmin_B - binsize/2 : binsize : tmax_B + binsize/2 # define the grid by
 the ebin edges

resampling_method_B = BinnedResampling(grid_B, 1000)

## standard version (1 kyr grid, with full uncertainties)
@time B_binned_fullength_ageunc = resample(uivD_B, resampling_method_B)


### Binned resampling on a 500 yr grid, for high resolution analysis of time ste
p 500 yrs with Grant, La2004, Lambert, Martinez-García

binsize_hr500 = 0.5 # mean resolution is one observation every 490 years
tmin_B = ceil(minimum(t_B))  # first bin midpoint at a whole kyr
tmax_B = floor(maximum(t_B)) # last bon midpoint at a whole kyr
grid_B_hr500 = tmin_B - binsize_hr500/2 : binsize_hr500 : tmax_B + binsize_hr500
/2 # define the grid by bin edges

resampling_method_B_hr500 = BinnedResampling(grid_B_hr500, 1000) # resample 1000
values within each bin of the grid.

@time B_binned_fullength_ageunc_hr500 = resample(uivD_B, resampling_method_B_hr5
00)
```

```
 31.571611 seconds (34.45 M allocations: 7.322 GiB, 8.51% gc time)
 28.384372 seconds (29.06 M allocations: 12.474 GiB, 13.74% gc time)

UndefVarError: resampling_method not defined

Stacktrace:
 [1] top-level scope at util.jl:156
 [2] top-level scope at In[13]:33
```

In [14]:

```
#### Binned resampling on a 125 yr grid, for high resolution analysis  of time s
tep 125 yrs with Grant, La2004,
binsize_hr125 = 0.125 # mean resolution is one observation every 490 years
tmin_B = ceil(minimum(t_B))  # first bin midpoint at a whole kyr
tmax_B = floor(maximum(t_B)) # last bon midpoint at a whole kyr
grid_B_hr125 = tmin_B - binsize_hr125/2 : binsize_hr125 : tmax_B + binsize_hr125
/2 # define the grid by bin edges

resampling_method_B_hr125 = BinnedResampling(grid_B_hr125, 1000) # resample 1000
values within each bin of the grid.
@time B_binned_fullength_ageunc_hr125 = resample(uivD_B, resampling_method_B_hr1
25)
```

```
 68.614789 seconds (64.92 M allocations: 46.878 GiB, 19.33% gc time)
```

Out[14]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 6409 uncertain values coupled with 6409 uncertain ind
ices
```

1. Binned resampling without age model uncertainty.

Version without age model uncertainties is for analysis with other EDC ice core dust record (Lambert / IceDust)

In [15]:

```
#### version without age model uncertainties, for analysis with other EDC ice co
re dust record (Lambert IceDust)

# binned on the 1 kyr grid
@time B_binned_fullength_noageuncEDC_noIntp = resample(uivD_B_EDC, resampling_me
thod_B)

# binned on the 500 yr grid
@time B_binned_fullength_noageuncEDC_hr500_noIntp = resample(uivD_B_EDC, resampl
ing_method_B_hr500)

#binned on the 125 yr grid
#@time B_binned_fullength_noageuncEDC_hr125_noIntp = resample(uivD_B_EDC, resamp
ling_method_B_hr125)
```

```
  6.276096 seconds (5.43 M allocations: 5.816 GiB, 24.78% gc time)
 12.876414 seconds (11.52 M allocations: 11.521 GiB, 25.87% gc time)
```

Out[15]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1603 uncertain values coupled with 1603 uncertain ind
ices
```

In [18]:

```
@time B_binned_fullength_noageuncEDC_hr125_noIntp = resample(uivD_B_EDC, resampl
ing_method_B_hr125)
```

```
45.891576 seconds (42.39 M allocations: 45.387 GiB, 25.86% gc time)
```

Out[18]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 6409 uncertain values coupled with 6409 uncertain ind
ices
```

In [19]:

```
@save "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
intp.jld2" B_binned_fullength_ageunc B_binned_fullength_ageunc_hr500 B_binned_fu
llength_ageunc_hr125 B_binned_fullength_noageuncEDC_noIntp B_binned_fullength_no
ageuncEDC_hr500_noIntp B_binned_fullength_noageuncEDC_hr125_noIntp
```

In [20]:

```
@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
intp.jld2"
```

Out[20]:

```
6-element Array{Symbol,1}:
 :B_binned_fullength_ageunc
 :B_binned_fullength_ageunc_hr500
 :B_binned_fullength_ageunc_hr125
 :B_binned_fullength_noageuncEDC_noIntp
 :B_binned_fullength_noageuncEDC_hr500_noIntp
 :B_binned_fullength_noageuncEDC_hr125_noIntp
```

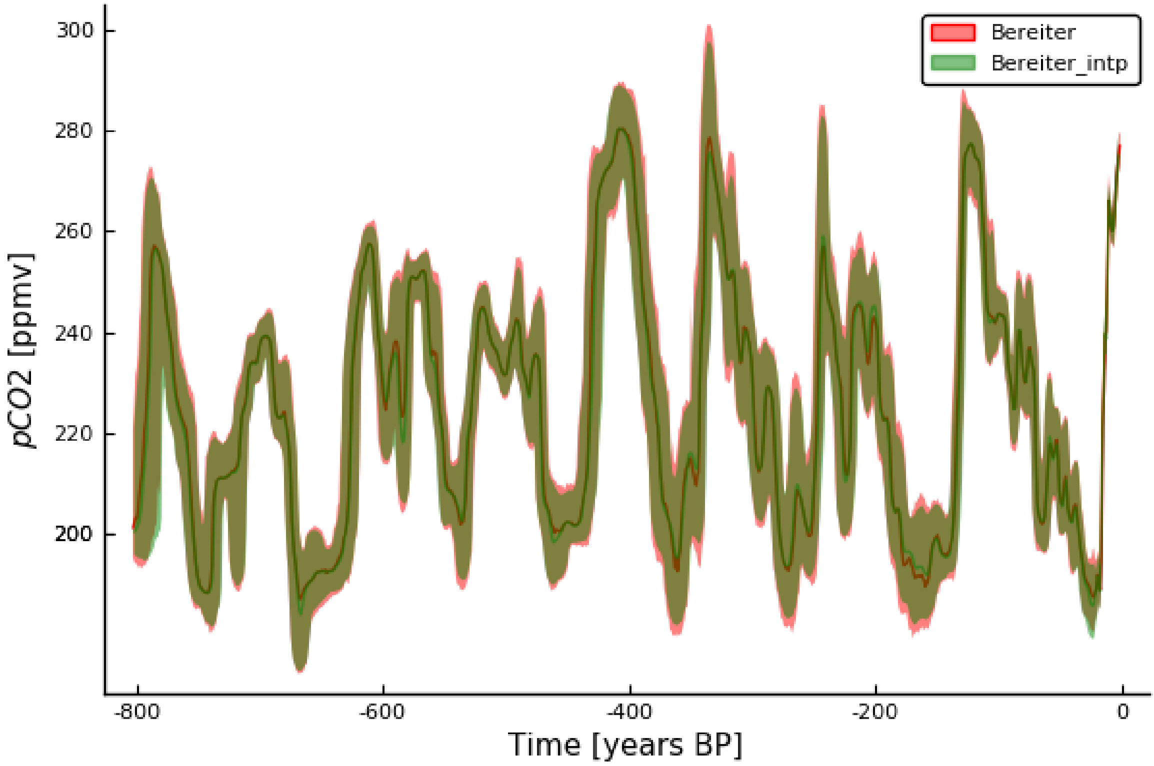
Plot the binned resampled time series with the 95% confidence interval

1. With the AICC2012 age model uncertainty

In [21]:

```
# 1 kyr grid, full age uncertainty,
############# no interpolation

@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
intp.jld2" # no interpolation
### Plot the binned resampled uivD time series with the 95% confidence interval
B = B_binned_fullength_ageunc

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

# time array
binmidpoints_B = [B.indices[i].value for i in 1:length(B)]


#plot_B_binned_ageunc =
plot(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :red,
    label = "Bereiter",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
    )
################ w/interpolation
@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter.jl
d2" # with interpolation
### Plot the binned resampled uivD time series with the 95% confidence interval
B = B_binned_fullength_ageunc

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

# time array
binmidpoints_B = [B.indices[i].value for i in 1:length(B)]


#plot_B_binned_ageunc_intp =
plot!(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :green, alpha = 0.5,
    label = "Bereiter_intp",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
    )
```

Out[21]:

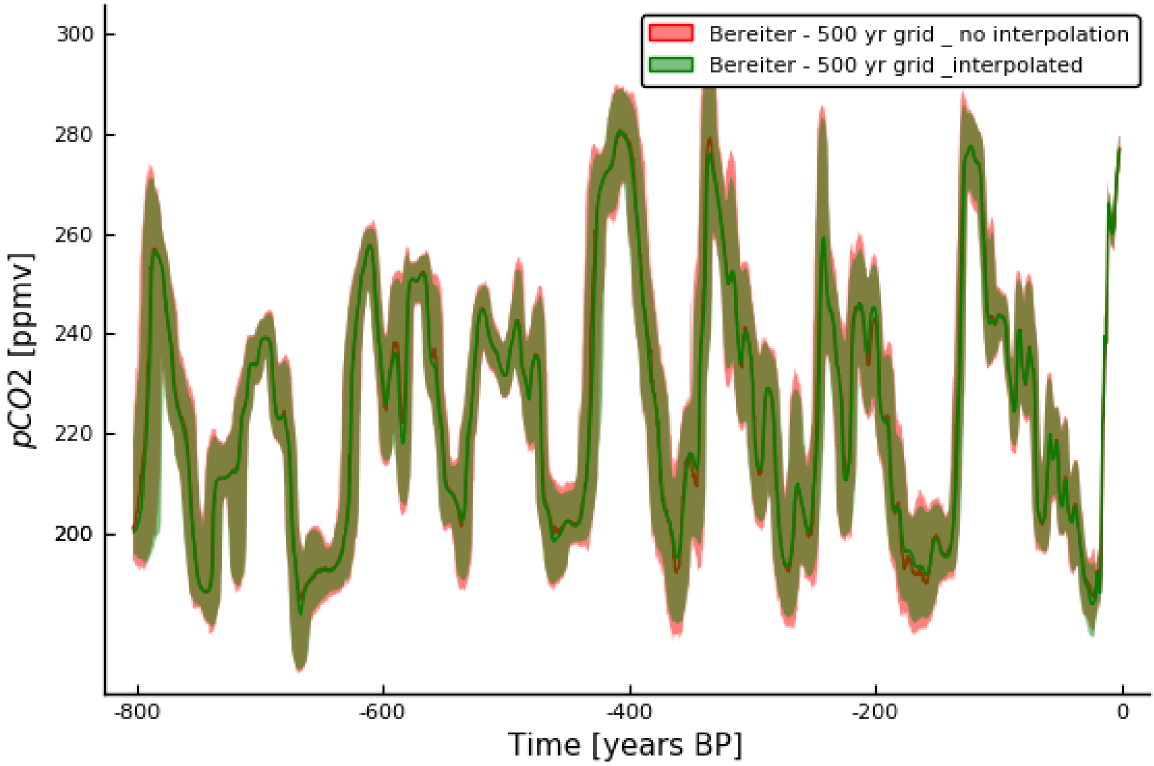


Bereiter
Bereiter_intp
pCO2 [ppmv]
300
280
260
240
220
200
-800
-600
-400
-200
0
Time [years BP]

In [117]:

```
# 500 yr grid, with age uncertainty

###### without interpolation,
@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
intp.jld2"
### same time series without age uncertainty
B = B_binned_fullength_ageunc_hr500

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

binmidpoints_B = [B.indices[i].value for i in 1:length(B)]# time array

plot(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :red,
    label = "Bereiter - 500 yr grid _ no interpolation",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
    )


######### with interpolation
@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter.jl
d2"

### same time series without age uncertainty
B = B_binned_fullength_ageunc_hr500

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

binmidpoints_B = [B.indices[i].value for i in 1:length(B)]# time array



plot!(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :green,
    label = "Bereiter - 500 yr grid _interpolated",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
    )
```

Out[117]:



These two versions give slightly different results in the predictive asymmetry analysis between Grant GSL and Bereiter CO2. Both give a mean predictive asymmetry from BerCO2 to GraSL, but

- with interpolation, confidence intervals overlap, leaving some chance (maybe 20%) that the predictive asymmetry is unsignificant.
- without interpolation, confidence intervals diverge completely, leaving an unambiguous indication of unidirectional coupling

2. Without age model uncertainty

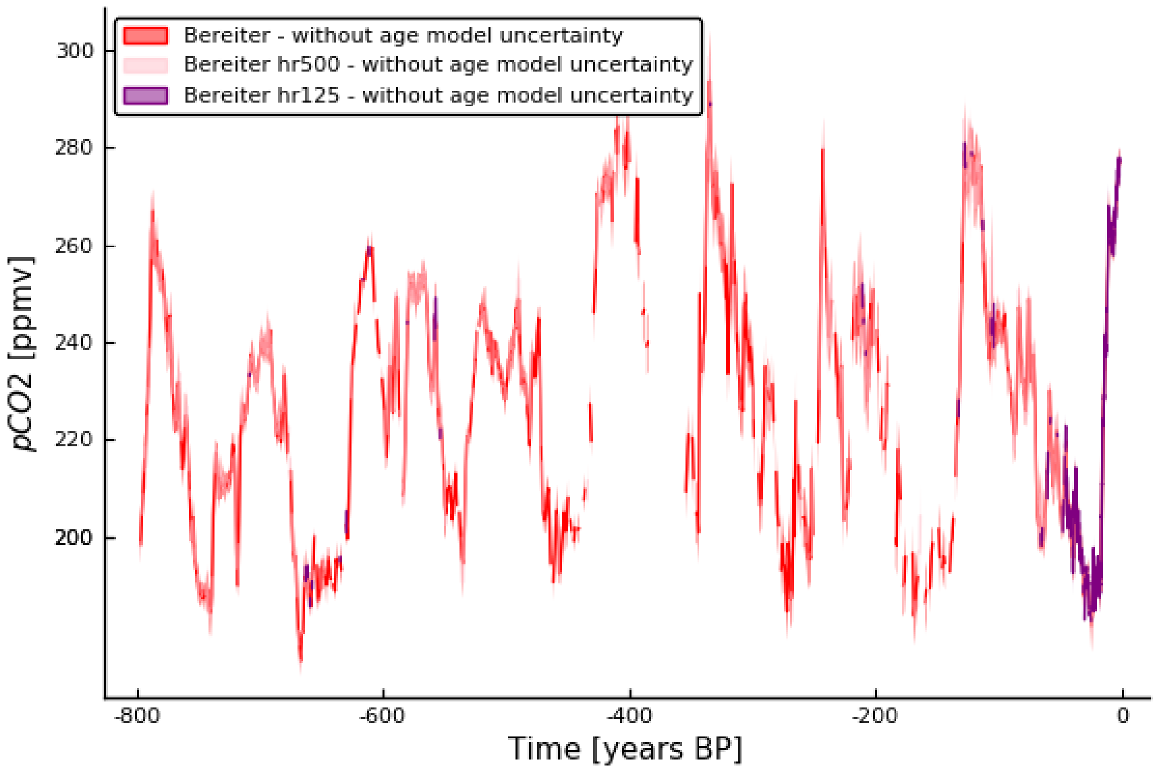
For analyses between EDC pCO2 and EDC dust - see that this calls for interpolation of data...

In [31]:

```
# 1 kyr, no age uncertainty
###### without interpolation
@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
intp.jld2"
B = B_binned_fullength_noageuncEDC_noIntp
# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(B.values, 0.5)
bin_upper = quantile.(B.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(B.values, 0.025)
# time array
binmidpoints_B = [B.indices[i].value for i in 1:length(B)]
#plot_B_binned_noageuncEDC_intp =
plot(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :red,
    label = "Bereiter - without age model uncertainty",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
    )

#### 500 yr grid

B = B_binned_fullength_noageuncEDC_hr500_noIntp
# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(B.values, 0.5)
bin_upper = quantile.(B.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(B.values, 0.025)
# time array
binmidpoints_B = [B.indices[i].value for i in 1:length(B)]
plot!(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :pink, alpha = 0.5,
    label = "Bereiter hr500 - without age model uncertainty",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
    )


#### 125 yr grid

B = B_binned_fullength_noageuncEDC_hr125_noIntp
# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(B.values, 0.5)
bin_upper = quantile.(B.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(B.values, 0.025)
# time array
binmidpoints_B = [B.indices[i].value for i in 1:length(B)]
plot!(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :purple,
    label = "Bereiter hr125 - without age model uncertainty",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
```

```
    )


######## NEED FOR INTERPOLATION
```

Out[31]:



> **We need to interpolate values, to run the analysis without age uncertainty.**

Interpolation of record when defined without age model uncertainty

Bereiter pCO2 and Lambert dust record are both from the EDC ice core. To better resolve the causal relationship between dust and pCO2, we can run analysis between these two records with the age uncertainty from lock-in depth of gas in ice, which is much smaller than the full age model uncertainty.

However, this calls for interpolation of some values on the Bereiter record, since there are gaps in the record that, when removing the age uncertainty, can no longer be covered in the binned resampling of data. To avoid over-interpolation, ee use the mean resolution of the record as the lower boundary for interpolation of data, which is 490 years for the Bereiter record.

In [179]:

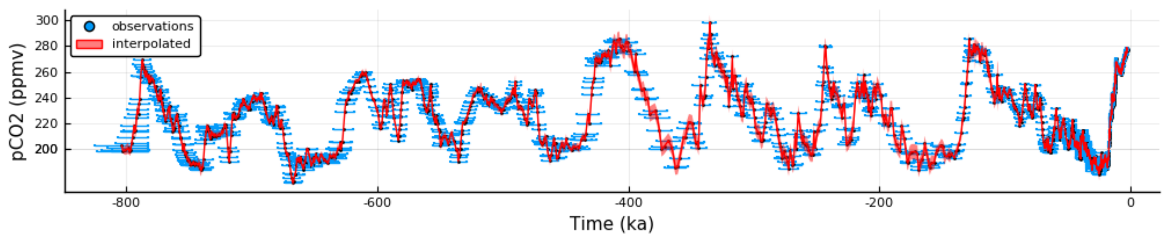
```
# interpolation of Bereiter data to ensure we have data in all bins

# defining interpolation functions
t_B_interpolate         = LinearInterpolation(t_B, t_B );
t_1σ_B_interpolate      = LinearInterpolation(t_1σ_B, t_B );
CO2_mean_B_interpolate = LinearInterpolation(CO2_mean_B , t_B );
CO2_1σ_B_interpolate    = LinearInterpolation(CO2_1σ_B , t_B );

# make a time grid to contain the interpolated values for the Bereiter dataset.
fine_grid_B = minimum(t_B) : 0.1 : maximum(t_B)  #### centennial resolution when
interpolated values are carried on
print(fine_grid_B)

# we then make arrays of the interpolated values corresponding to the bins of th
e fine time grid
intpD_t_B         = [t_B_interpolate(i) for i in fine_grid_B]
intpD_t_1σ_B      = [t_1σ_B_interpolate(i) for i in fine_grid_B];
intpD_CO2_mean_B = [CO2_mean_B_interpolate(i) for i in fine_grid_B];
intpD_CO2_1σ_B    = [CO2_1σ_B_interpolate(i) for i in fine_grid_B];
;
```

```
-803.70925:0.1:-2.00925
```

In [180]:

```
# plot interpolation to control

# NOTE: COMPUTATIONALLY HEAVY CELL when plotting interpolated age uncertainties

plot_intpD_B =
plot(#title = "Bereiter interpolation on fine grid",
    xlabel = "Time (ka)",
    ylabel = "pCO2 (ppmv)",
    size = (1000,200))
scatter!(t_B, CO2_mean_B,
    #ribbon = (CO2_1σ_B, CO2_1σ_B),
    xerr = 2 * t_1σ_B,
    ms = 1,
    label = "observations")
plot!(intpD_t_B, intpD_CO2_mean_B,
    ribbon = (2 * intpD_CO2_1σ_B),
    color = "red",
    label = "interpolated")
#plot!(fine_grid_B, interpD_CO2_mean_B,
    # xerr = interpD_t_2σ_B, # plotting xerr is computationally heavy, and drown
s out plot info. Better to plot on "observations"
    # ms = 0.1,
    # label = "age uncertainty",
    # color = "red")
```

Out[180]:



Define the interpolated data as an uivD

In [181]:

```
# uivD_B_EDC defined without age uncertainty, for analysis with other records fr
om the EDC core.

t_uiv_B_EDC = [UncertainValue(Normal, intpD_t_B[i], 0) for i in 1:length(intpD_t
_B)] # age model uncertainty not included
CO2_uiv_B = [UncertainValue(Normal, intpD_CO2_mean_B[i], intpD_CO2_1σ_B[i]) for
i in 1:length(intpD_CO2_mean_B)]
uivD_B_EDC = UncertainIndexValueDataset(t_uiv_B_EDC, CO2_uiv_B)

# Save the relevant arrays of the Bereiter record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Bereiter_noageuncEDC_int
p.jld2" uivD_B_EDC
```

In [174]:

```
length(t_uiv_B_EDC)
```

Out[174]:

```
1604
```

In [184]:

```
mean_res = mean(diff(t_B))
print("To limit the effect of false values, we do not interpolate values below t
he mean resolution of the record (", mean_res, " kyrs).
The highest resolution analyses with th is record will therefore be with the 500
yr time step.")
```

```
To limit the effect of false values, we do not interpolate values be
low the mean resolution of the record (0.48825185140073074 kyrs).
The highest resolution analyses with th is record will therefore be
with the 500 yr time step.
```

- Binned resampling on the 1 kyr grid

In [185]:

```
# version without age model uncertainties, for analysis with other EDC ice core
 record (Bereiter pCO2)

# define the 1 kyr grid
binsize = 1 # 1 kyr
tmin_B = ceil(minimum(t_B))   # first bin midpoint at a whole kyr
tmax_B = floor(maximum(t_B))  # last bon midpoint at a whole kyr
grid_B = tmin_B - binsize/2 : binsize : tmax_B + binsize/2 # define the grid by
 the bin edges


# binned resamplig on the 1 kyr grid
resampling_method = BinnedResampling(grid_B, 1000) # draw 1000 samples in each b
in (uncertainties are now only in value)
@time B_binned_fullength_noageuncEDC = resample(uivD_B_EDC, resampling_method_B)
```

```
 32.186467 seconds (26.09 M allocations: 28.265 GiB, 29.33% gc time)
```

Out[185]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 802 uncertain values coupled with 802 uncertain indic
es
```

- Binned resampling on a higher resolution time grid (one value for every 500 years)

In [186]:

```
# Binned resampling on a 500 yr grid, for high resolution analysis with Lambert
 EDC record (low age uncertainty)

binsize_hr500 = 0.5 # mean resolution is one observation every 490 years
tmin_B = ceil(minimum(t_B))  # first bin midpoint at a whole kyr
tmax_B = floor(maximum(t_B)) # last bon midpoint at a whole kyr
grid_B_hr500 = tmin_B - binsize_hr500/2 : binsize_hr500 : tmax_B + binsize_hr500
/2 # define the grid by bin edges

# binned resampling on the 500 yr grid
resampling_method_B_hr500 = BinnedResampling(grid_B_hr500, 1000) # resample 1000
values within each bin of the grid.
@time B_binned_fullength_noageuncEDC_hr500 = resample(uivD_B_EDC, resampling_met
hod_B_hr500)
```

```
 73.369700 seconds (51.91 M allocations: 55.877 GiB, 26.66% gc time)
```

Out[186]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1603 uncertain values coupled with 1603 uncertain ind
ices
```

In [187]:

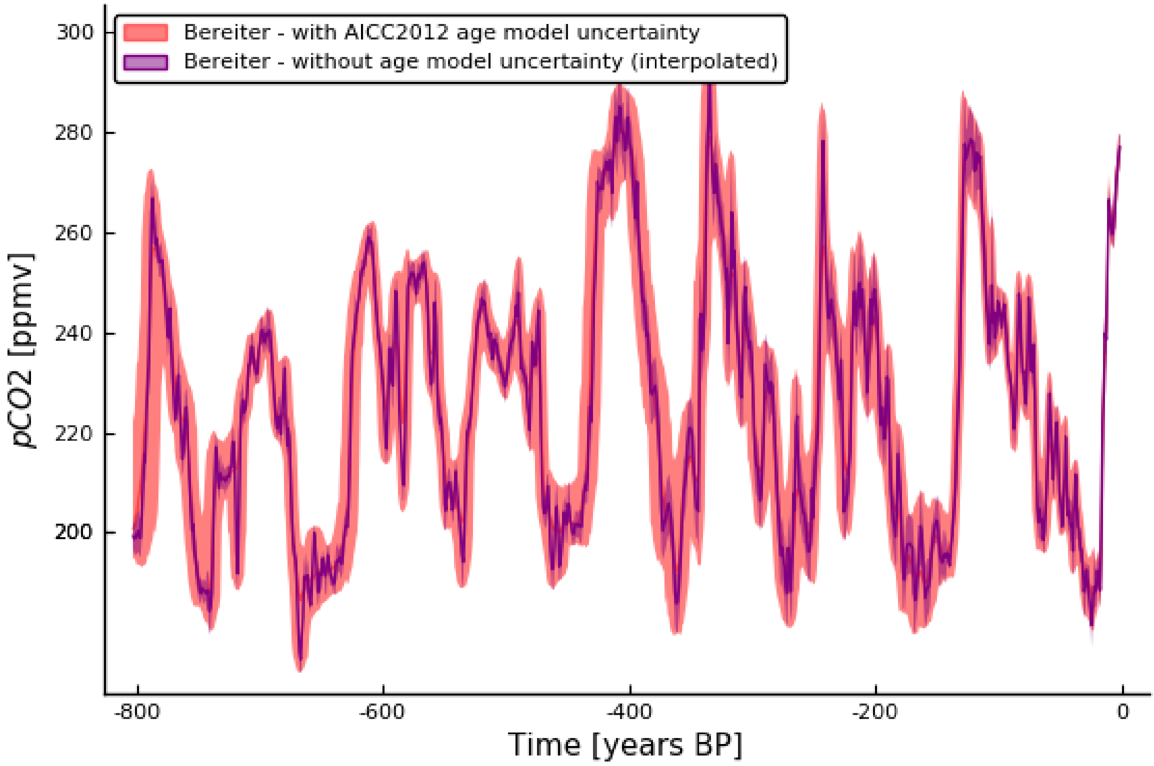
```
# Save the relevant arrays of the Bereiter record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
ageuncEDC_intp.jld2" B_binned_fullength_noageuncEDC B_binned_fullength_noageuncE
DC_hr500
```

- Visualize the difference between the records with and without age model uncertainty

In [159]:

```
# 1 kyr grid, full age uncertainty,
############# no interpolation

@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
intp.jld2" # no interpolation
### Plot the binned resampled uivD time series with the 95% confidence interval
B = B_binned_fullength_ageunc

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

# time array
binmidpoints_B = [B.indices[i].value for i in 1:length(B)]


#plot_B_binned_ageunc =
plot(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :red, alpha = 0.5,
    label = "Bereiter - with AICC2012 age model uncertainty",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false
    )
################ w/interpolation
@load "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/Bereiter_no
ageuncEDC_intp.jld2" #with interpolation
### Plot the binned resampled uivD time series with the 95% confidence interval
B = B_binned_fullength_noageuncEDC

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

# time array
binmidpoints_B = [B.indices[i].value for i in 1:length(B)]


#plot_B_binned_ageunc_intp =
plot!(binmidpoints_B, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :purple, alpha = 1,
    label = "Bereiter - without age model uncertainty (interpolated)",
    xlabel = "Time [years BP]",
    ylabel = string(L"pCO2", " [ppmv]"),
    grid = false, legend = :topleft
    )
```

Out[159]:



4.2 - Chalk pCO2 record

- pCO2 record from *Chalk et al. (2017)*, denoted **C**
- d11B proxy record for pCO2 spanning 1.090 - 1.240 Ma, (this is the only high resolution pCO2 record that is synchronous to the MPT).
- Data available from...DOI...

i) Read in data

Age [ka BP] CO2 [ppmv] (median CO2 (ppm)) CO2 [ppmv] CO2 [ppmv] CO2 [ppmv] (probablistic assessment of CO...) CO2 [ppmv] (probability maximum 95% lower...) CO2 [ppmv] (probability maximum 95% upper...)Parameter(s): AGE [ka BP] (Age) Carbon dioxide [ppmv] (CO2) # COMMENT: median CO2 (ppm) Carbon dioxide [ppmv] (CO2) # COMMENT: CO2 (ppm) probabilistic upper bound of 95% probability interval [sic.] Carbon dioxide [ppmv] (CO2) # COMMENT: CO2 (ppm) probabilistic upper bound of 95% probability interval Carbon dioxide [ppmv] (CO2) # COMMENT: probablistic assessment of CO2 (ppm) Carbon dioxide [ppmv] (CO2) # COMMENT: probability maximum 95% lower bound for mean (ppm) Carbon dioxide [ppmv] (CO2) # COMMENT: probability maximum 95% upper bound for mean (ppm)

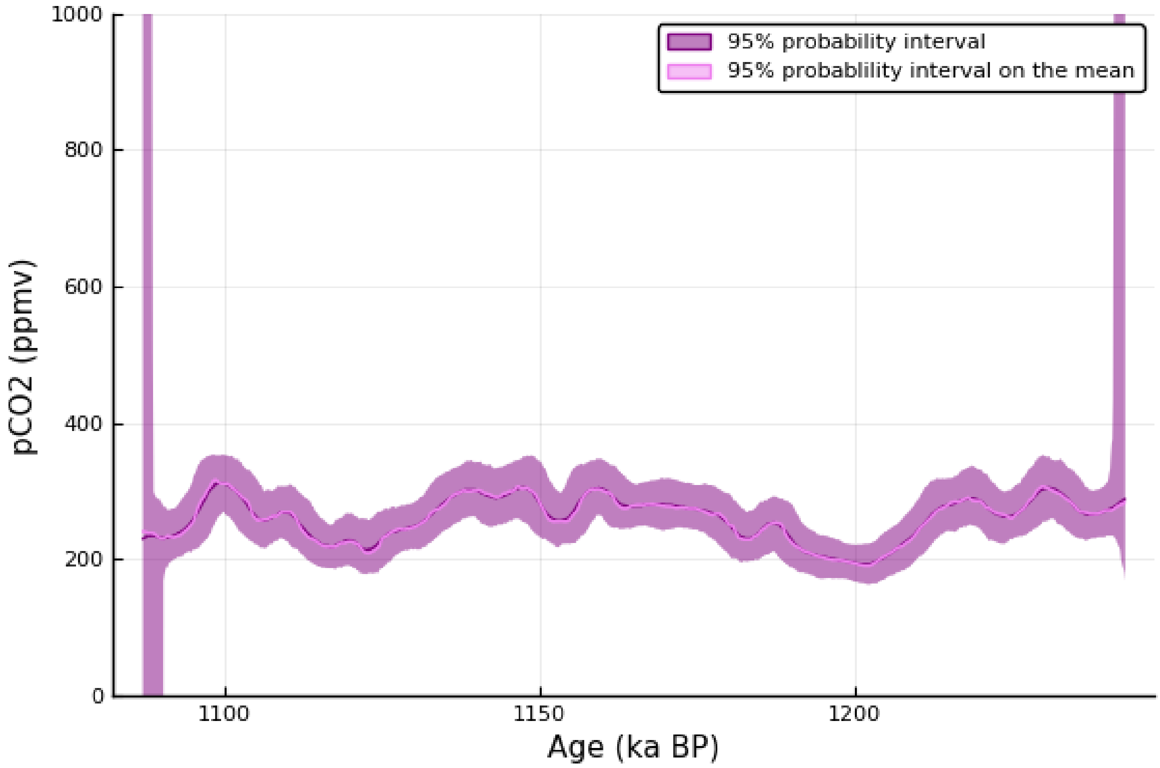
In [186]:

```
# CO2 Chalk (d11B proxy data)
data_CO2_Chalk = DelimitedFiles.readdlm("../../MASTER_2.0/data/CO2/Chalk2017_CO
2.tab", Float64, skipstart = 18);

#Chalk et al dataset reports the following:

# 95% confidence interval
age_C = data_CO2_Chalk[:,1];       # Age (ka BP)

CO2o_mean_C = data_CO2_Chalk[:,2]; # median CO2 (ppmv)
CO2o_lo_C = data_CO2_Chalk[:,3]    # CO2 (ppm) probabilistic upper bound of 95%
 probability interval - [sic.] - probably mean lower bound]
CO2o_up_C = data_CO2_Chalk[:,4]    # CO2 (ppm) probabilistic upper bound of 95%
 probability interval

# 95% confidence interval on the mean
CO2p_mean_C = data_CO2_Chalk[:,5]; # probablistic assessment of CO2 (ppm)
CO2p_lo_C = data_CO2_Chalk[:,6]    # probability maximum 95% lower bound for mea
n (ppmv)
CO2p_up_C = data_CO2_Chalk[:,7]    # probability maximum 95% upper bound for mea
n (ppmv)
;
```

In [187]:

```
# plot Chalk
plot_Chalk =
plot(#title = "syn-MPT pCO2 record (Chalk)",
    xlabel = "Age (ka BP)",
    ylabel = "pCO2 (ppmv)",
    xticks = (:1050:50:1250),
    ylims = (0,1000)
)
plot!(age_C, CO2o_mean_C,
    ribbon = (CO2o_mean_C .- CO2o_lo_C, CO2o_up_C .- CO2o_mean_C),
    label = "95% probability interval",
    color = :purple)
plot!(age_C, CO2p_mean_C,
    ribbon = (CO2p_mean_C .- CO2p_lo_C, CO2p_up_C .- CO2p_mean_C),
    label = "95% probablility interval on the mean",
    color = :violet)
```

Out[187]:



To fully represent the uncertainty, we will use the 95% probability interval (purple ribbon in plot above).

However, there is obviously something off with the first and last datapoints reported in the Chalk dataset (a range from -10^5 ppmv to 5*10^4 ppmv is non-sensical). We therefore will not include the first and last datapoints of the record, where values deviate from the range in the remainder of the record. We have prepared a second version of the .tab-file, where we have cut out the improbable values (first 3 kyr and last 2 kyrs of observations)

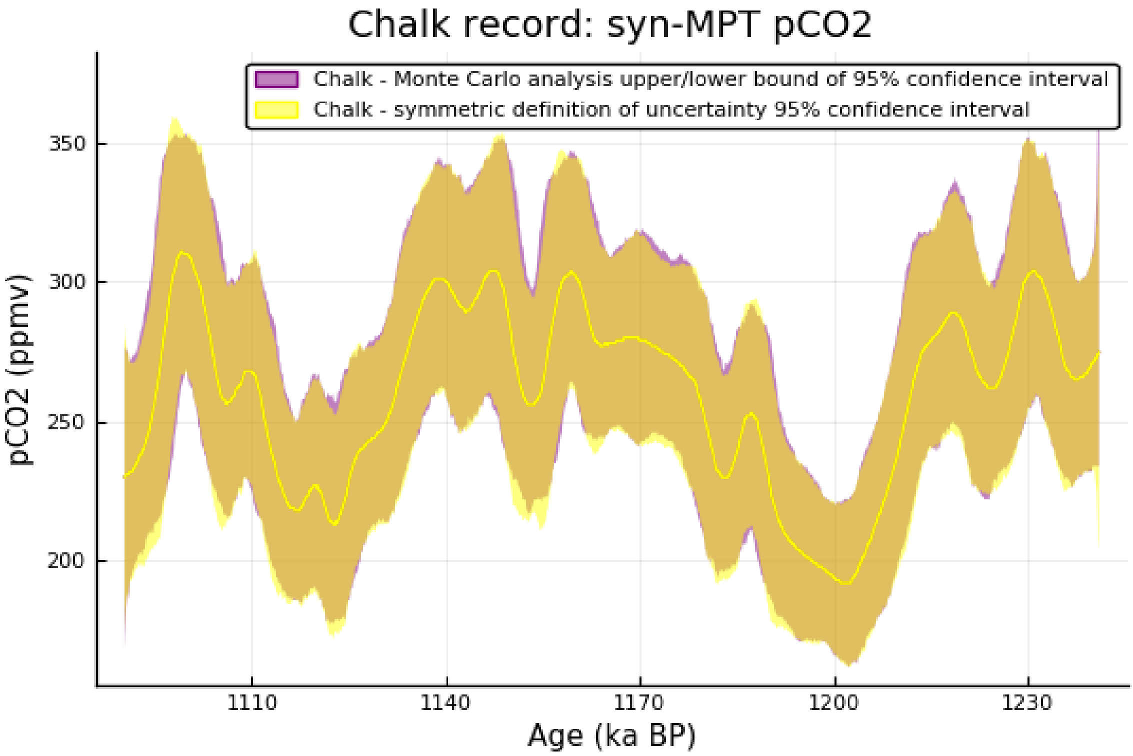
In [40]:

```
# data_CO2_Chalk[ (age_C .> 1090) .& (age_C .< 1241) ] # Didn't work to cut this way, so we cut directly in the data file
```

The pCO2 record we will use is plotted below:

In [188]:

```
# CO2 Chalk (d11B proxy data)
data_CO2_Chalk = DelimitedFiles.readdlm("../../MASTER_2.0/data/CO2/Chalk2017_CO2
_v2.tab", Float64, skipstart = 18); # We have cut out the time interva


age_C_ = data_CO2_Chalk[:,1];       # Age (ka BP)
CO2_mean_C_ = data_CO2_Chalk[:,2]; # median CO2 (ppmv)
CO2_lo_C_ = data_CO2_Chalk[:,3]    # CO2 (ppm) probabilistic lower bound of 95%
 probability interval
CO2_up_C_ = data_CO2_Chalk[:,4]    # CO2 (ppm) probabilistic upper bound of 95%
 probability interval

# defining an array for standard deviation, needed when defining our uivD
CO2_1σ_C_   = (CO2_up_C_ .- CO2_lo_C_) / 4     # St.dev (1σ) equals 1/4 of the 9
5% CI (2σ on each side of mean)input when defining the uivD

plot_Chalk_age =
plot(title = "Chalk record: syn-MPT pCO2",
    xlabel = "Age (ka BP)",
    ylabel = "pCO2 (ppmv)")
plot!(age_C_, CO2_mean_C_,
    ribbon = (CO2_mean_C_ .- CO2_lo_C_, CO2_up_C_ .- CO2_mean_C_),
    label = "Chalk - Monte Carlo analysis upper/lower bound of 95% confidence in
terval",
    color = :purple)
plot!(age_C_, CO2_mean_C_,
    ribbon = (2*CO2_1σ_C_, 2*CO2_1σ_C_),
    label = "Chalk - symmetric definition of uncertainty 95% confidence interva
l",
    color = "yellow")

#=  There are minor variations in the two ways of defining the 95% confidence in
terval.
This is due to asymmetry in upper and lower bounds
compared to a completely smooth hypothetical normal distribution.
We deem the differences to be so small that we can safely assume a normal distri
bution for the data. =#

savefig("../figurar/RawData/pCO2/Chalk_rawD_age.pdf")
```

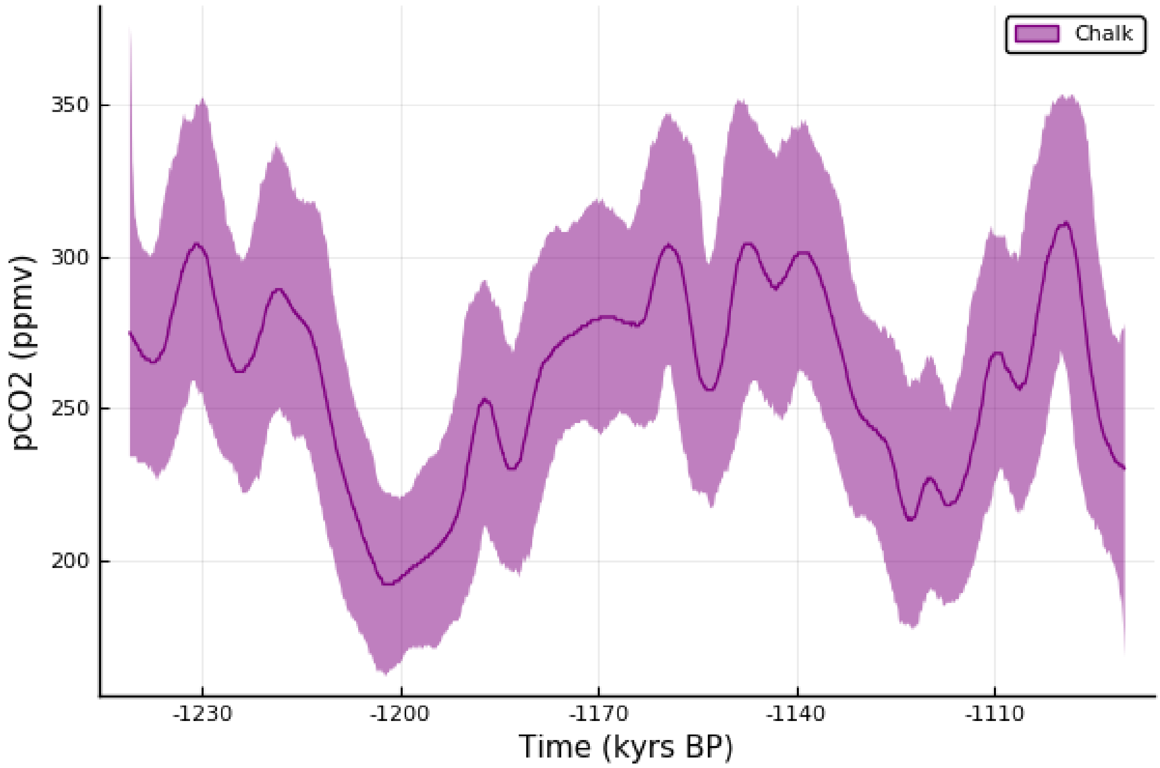


ii) Reverse arrays (from age to time)

In [189]:

```
# We reverse the arrays we will use to get the dataset of time (running forward
s)
t_C = -reverse(age_C_, dims = 1)
CO2_mean_C = reverse(CO2_mean_C_, dims = 1)
CO2_lo_C   = reverse(CO2_lo_C_, dims = 1)
CO2_up_C   = reverse(CO2_up_C_, dims = 1)
# defining an array for standard deviation, needed when defining our uivD
CO2_1σ_C   = (CO2_up_C .- CO2_lo_C) / 4    # St.dev (1σ) equals 1/4 of the 95% C
I (2σ on each side of mean)input when defining the uivD

@save "../Koding/WrangledDataFiles/BasicArrays/Chalk.jld2" t_C CO2_mean_C CO2_lo
_C CO2_up_C CO2_1σ_C


# check by plotting
plot_Chalk =
plot(xlabel = "Time (kyrs BP)",
    ylabel = "pCO2 (ppmv)")
plot!(t_C, CO2_mean_C,
    ribbon = (CO2_mean_C .- CO2_lo_C, CO2_up_C .- CO2_mean_C),
    color = :purple,
    label = "Chalk")
```

Out[189]:



iii) Age model uncertainty

The age model of the record comes from aligning the benthic foram $\delta^{18}O$ record from the same core (ODP999) to the the LR04 reference stack. [Check: it is unclear whether or not the age model uncertainty was incorporate in the Monte Carlo anaysis of the by Chalk et al.] We therefore make an array add the LR04 age model uncertainty (95% confidence envelope = 6 kyrs for the Chalk time interval gives 1\sigma = 1.5 kyrs).

In [191]:

```
# age model uncertainty from LR04
t_1σ_C = [interpolate_t_1σ_LR04(i) for i in t_C]
;
```

```
UndefVarError: interpolate_t_1σ_LR04 not defined

Stacktrace:
 [1] top-level scope at In[191]:1
```

Since our method operates with a lead-lag definition of causality, it is very sensitive to age reversals, and it is crucial to include uncertainties in the time series' age models.

The age model was constructed from a benthic $\delta^{18}O$ record from the same site, which has been tuned to the LR04 reference age model. We therefore add the LR04 age model uncertainty to the index uncertainty when redefining the Chalk record datatype to uivD.

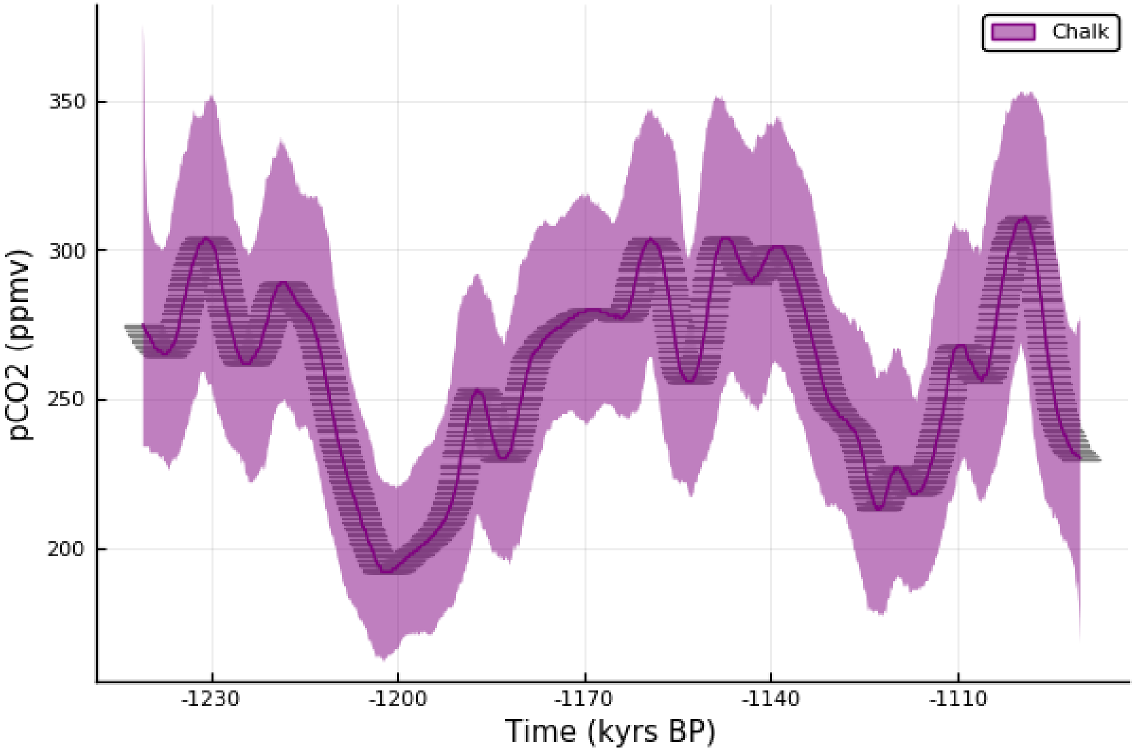
The LR04 age model full uncertainty enveolpe uncertainty reported for 1-3 Ma BP is 6 kyrs (Lisiecki & Raymo, 2005). We interpreted this as the 95% confidence interval, aka 4σ. This gives us a 1σ age uncertainty value of 1.5 kyr for the entire span of the Chalk record.

In [194]:

```
# Create an array for age model uncertainty for the Chalk record

# t_1σ_LR04[t_C] ;
# Doesn't work to redefine LR04 age model uncertainty this way, probably due to
 dimension mismatch.

# We therefore create an empty array and fill it with the corresponding value fo
r 1σ
t_1σ_C = zeros(length(t_C))
for i in 1:length(t_1σ_C)
    t_1σ_C[i] = 1.5 # 1 sigma = 1 kyr for the entire length of the Chalk time se
res
end
t_1σ_C; # Check, all 1.5 = all good.
```

In [195]:

```
# check by plotting
plot_Chalk =
plot(xlabel = "Time (kyrs BP)",
    ylabel = "pCO2 (ppmv)")
plot!(t_C, CO2_mean_C, xerr = 2*t_1σ_C, color = :grey, ms = 0.1, label = "")
plot!(t_C, CO2_mean_C,
    ribbon = (CO2_mean_C .- CO2_lo_C, CO2_up_C .- CO2_mean_C),
    color = :purple,
    label = "Chalk")


savefig("../figurar/RawData/pCO2/plot_Chalk_rawD_timeuncLR04.pdf")
```



In [196]:

```
@save "../Koding/WrangledDataFiles/BasicArrays/Chalk.jld2" t_C t_1σ_C CO2_mean_C
CO2_lo_C CO2_up_C CO2_1σ_C
```

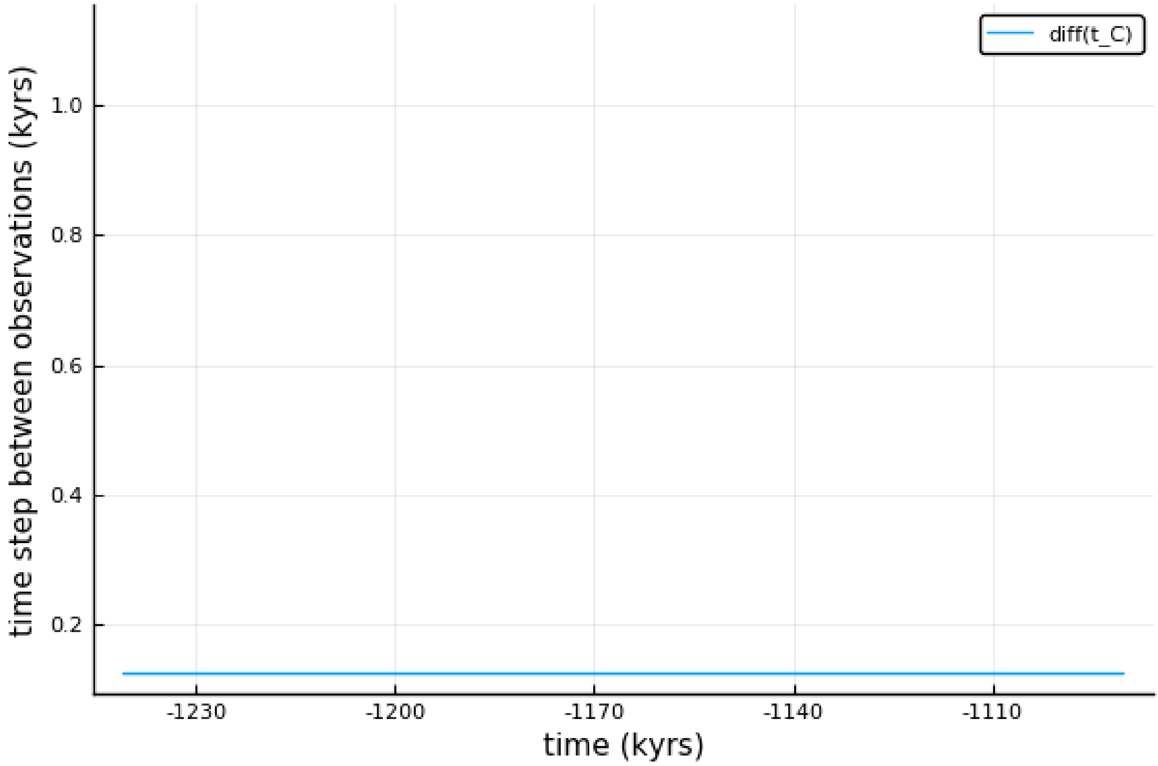
Resolution

We now check the temporal resolution and evt need for interpolation on the Chalk dataset:

In [73]:

```
# check temporal resolution on the Chalk dataset

mean(diff(t_C))     # 125 years
minimum(diff(t_C)) # 125 years
maximum(diff(t_C)) # 125 years

plot(t_C, diff(t_C),
    xlabel = "time (kyrs)",
    ylabel = "time step between observations (kyrs)",
    label = "diff(t_C)")
```

Out[73]:



The Chalk dataset is a high resolution record, requiring no interpolation. Observations are on a regular grid, with one observation for every 125 years, same as Rohling data, making it possible to do a more high-resolution analysis between these two time series.

Important note on resolution: These are not the raw data. Chalk et al. (2017) report original samples for every 3.5 - 4.5 years (Supporting information). The resolution they publish in the dataset is thus an order of magnitude above the sampled resolution. The over-sampling of the data may bias the results, as we use a data-driven method.

v) Redefining Chalk data as an `UncertainIndexValueDataset` **

In [21]:

```
# Redefining Chalk as uivD
t_uiv_C = [UncertainValue(Normal, t_C[i], t_1σ_C[i]) for i in 1:length(t_C)]
CO2_uiv_C = [UncertainValue(Normal, CO2_mean_C[i], CO2_1σ_C[i]) for i in 1: leng
th(CO2_mean_C)]
uivD_C = UncertainIndexValueDataset(t_uiv_C, CO2_uiv_C)
```

Out[21]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1207 uncertain values coupled with 1207 uncertain ind
ices
```

In [23]:

```
# plot(uivD_C)
```

In [24]:

```
# Save the relevant arrays of the Chalk record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Chalk.jld2" uivD_C
```

vi) Binned resampling

In [25]:

```
@load "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Chalk.jld2"
```

Out[25]:

```
1-element Array{Symbol,1}:
 :uivD_C
```

> Due to a false positive (with La2004 on 125 kyr timestep grid) we wish to use this record for sensitivity analysis. We therefore prepare many versions of the time series binned with different resolutions.
>
> - standard 1 kyr timestep grid

In [29]:

```
# make 1 kyr timestep grid
binsize = 1
tmin_C = ceil(minimum(t_C))
tmax_C = floor(maximum(t_C))
grid_C = tmin_C + binsize/2 : binsize : tmax_C - binsize/2

resampling_method_C = BinnedResampling(grid_C,1000)
```

Out[29]:

```
BinnedResampling{StepRangeLen{Float64,Base.TwicePrecision{Float64},B
ase.TwicePrecision{Float64}}}(-1240.5:1.0:-1091.5, 1000)
```

In [30]:

```
@time C_binned = resample(uivD_C, resampling_method_C)
```

```
 12.806340 seconds (16.77 M allocations: 1.680 GiB, 4.88% gc time)
```

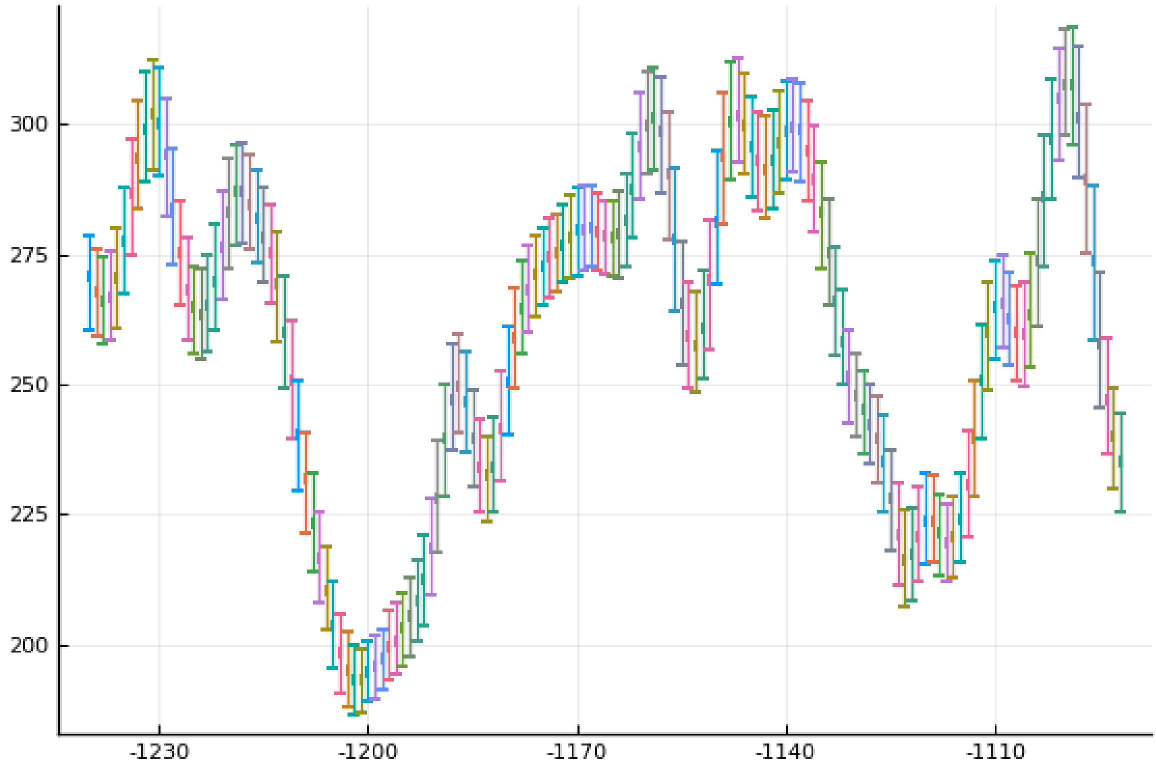
Out[30]:

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

In [118]:

```
p1 = plot(C_binned)
#xgrid!(p1, :on, :red, 1, :dash)
```

Out[118]:



Make a high resolution version of the time series, for high resolution analyses.

- `C_binned_hr125` with time step of 125 years, for hr analysis with La2004

In [31]:

```
binsize_hr = 0.125
grid_C_hr = tmin_C + binsize_hr/2 : binsize_hr : tmax_C - binsize_hr/2

resampling_method_C_hr = BinnedResampling(grid_C_hr, 1000)

@time C_binned_hr0125 = resample(uivD_C, resampling_method_C_hr)
```

```
 17.373565 seconds (21.50 M allocations: 7.177 GiB, 13.13% gc time)
```

Out[31]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1199 uncertain values coupled with 1199 uncertain ind
ices
```

- `C_binned_hr500` with time step of 500 years, for hr analysis with Martinez-García

In [35]:

```
tmin_C = t_C[1]
tmax_C = t_C[end]
binsize_hr = 0.5
grid_C_hr500 = tmin_C + binsize_hr/2 : binsize_hr : tmax_C - binsize_hr/2
```

Out[35]:

```
-1240.75:0.5:-1090.75
```

In [34]:

```
resampling_method_C_hr500 = BinnedResampling(grid_C_hr500, 1000)

@time C_binned_hr500 = resample(uivD_C, resampling_method_C_hr500)
```

```
 12.953482 seconds (16.60 M allocations: 2.432 GiB, 5.52% gc time)
```

Out[34]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 300 uncertain values coupled with 300 uncertain indic
es
```

vii) Save the binnedresampled timeseries

In [36]:

```
@save "../Koding/WrangledDataFiles/Binned_ts_fullength/Chalk.jld2" C_binned C_bi
nned_hr0125 C_binned_hr500 # uivD_C
```

In [37]:

```
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/Chalk.jld2"
```

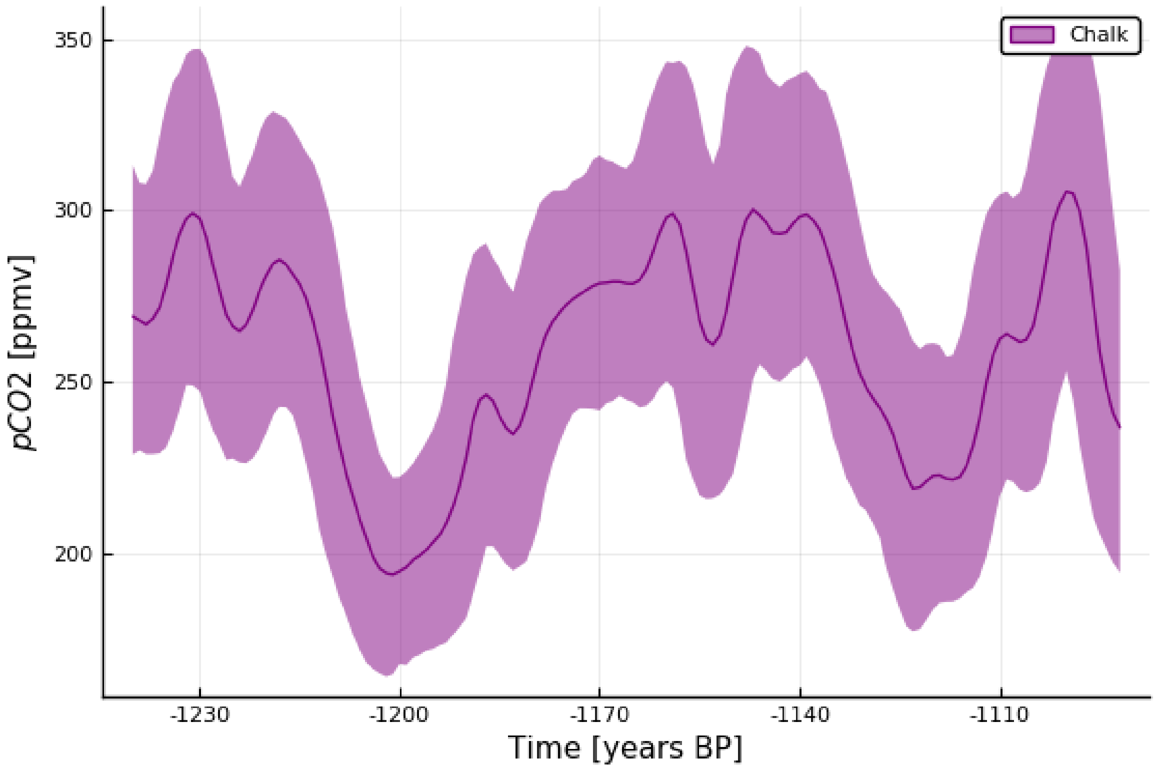
Out[37]:

```
3-element Array{Symbol,1}:
 :C_binned
 :C_binned_hr0125
 :C_binned_hr500
```

In [38]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
C = C_binned

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

# time array
binmidpoints_C = [C.indices[i].value for i in 1:length(C)]


plot_C_binned =
plot(binmidpoints_C, bin_median,
     ribbon = (bin_lower, bin_upper),
     color = :purple,
     label = "Chalk",
     xlabel = "Time [years BP]",
     ylabel = string(L"pCO2", " [ppmv]"),
     #grid = false
     )
```

Out[38]:



4.3 - Hönisch pCO2 record

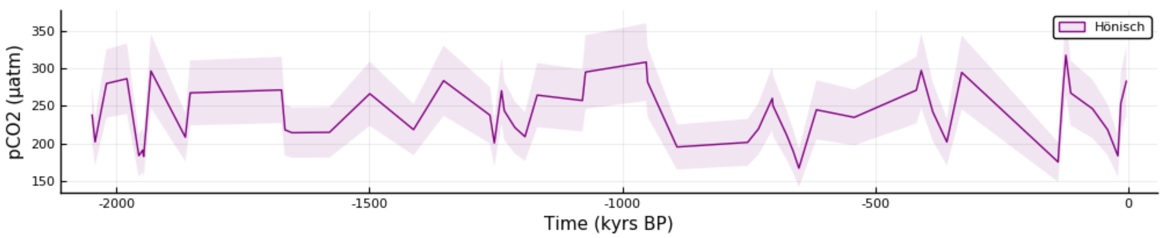
- From *Hönisch et al. (2009)*, denoted **H**
- Low-resolution pCO2 record from d11B proxy, spanning ca 2 Ma.
- Data available from ... DOI:..

Note: This is a very low resolution (only 54 datapoints over 2 Ma). Therefore contains little information on the dynamics of pCO2. It will therefore not be meaningful to run our analysis, but it is nice to have to get an idea of range of pCO2 over time.

In [40]:

```
# CO2 Hönisch (2009)

rawD_Honisch = readdlm("../../MASTER_2.0/data/CO2/Honisch2009_data_v2.txt", '\t'
, Float64, '\r', dims = (54, 3))
# Note: have selected columns A, Q and R (specs below) from the pangaea file, an
d replaced empty cells with NaN.
revD_H = reverse(rawD_Honisch, dims = 1)

t_H = - revD_H[:,1] # A - redefining from age (ka BP) to time (kyrs) by making n
egative
CO2_H = revD_H[:,2]        # Q - pCO2 [μatm] from calculations based on the varyi
ng alkalinity scenario (4%)
CO2_1σ_H = revD_H[:,3]     # R - standard deviation of pCO2
;

# plot
plot_Honisch =
plot(xlabel = "Time (kyrs BP)",
    ylabel = "pCO2 (μatm)",
    size = (1000,200))
plot!(t_H, CO2_H,
    color = :purple,
    fillalpha = 0.1,
    label = "Hönisch",
    #yerr = (2*CO2_1σ_H), # vertical error bars
    ribbon = (2*CO2_1σ_H, 2*CO2_1σ_H), # ± 2σ
    markerstrokecolor = :black,
    ms = 1,)
```

Out[40]:



In [41]:

```
savefig("../../Master_2.0/figurar/RawData/pCO2/plot_Honisch_ribbonscatter.pdf")
```

Now let's make a comparative plot of the pCO2 records

In [186]:

```
# comparative plot of pCO2 records

plot_pCO2_comparative =
plot(xlabel = "Time (kyrs BP)",
    ylabel = "pCO2 (ppmv)",
    size = (1000,400),
    legend = :topleft
    )

# Chalk:
plot!(t_C, CO2_mean_C,
    ribbon = (2 * CO2_1σ_C), # plotting 2σ to illustrate that we will carry on a
95% confidence interval
    color = :violet,
    label = "Chalk")

# Bereiter:
plot!(t_B, CO2_mean_B, xerr = (2 * t_1σ_B), ms = 0.1, color = :black,
    label = "2σ age uncertainty" # how can I remove this label completely?
    )
plot!(t_B, CO2_mean_B,
    ribbon = (2 * CO2_1σ_B),  # 95% CI
    color = :red,
    fillalpha = 0.5,
    label = "Bereiter")
#savefig("../../Master_2.0/figurar/RawData/pCO2/plot_pCO2_comparative_B_C_ageun
c.pdf")

# Honisch:
twinx()
ylabel = "pCO2 (μatm)"  # NEED HELP HERE on how to plot a second y-axis
scatter!(t_H, CO2_H,
    color = :purple,
    fillalpha = 0.1,
    label = "Hönisch",
    #yerr = (2*CO2_1σ_H), # vertical error bars
    ribbon = (2*CO2_1σ_H, 2*CO2_1σ_H), # ± 2σ
    markerstrokecolor = :purple,
    ms = 1
    )
#savefig("../../Master_2.0/figurar/RawData/pCO2/plot_pCO2_comparative_wHonisch.p
df")
```

```
UndefVarError: CO2_1σ_H not defined

Stacktrace:
 [1] top-level scope at In[186]:30
```

In [187]:

```
# overview plot of pCO2 records

# overview plot of pCO2 records

plot_pCO2_overview_B_C =
plot(plot_Bereiter,
    plot_Chalk,
    plot_Honisch,
    layout = grid(3,1),
    link = :x,
    size = (1000,400))
#savefig("../../Master_2.0/figurar/RawData/pCO2/plot_pCO2_overview_wHonisch.pd
f")

plot_pCO2_overview_B_C =
plot(plot_Bereiter,
    plot_Chalk,
    #plot_Honisch,
    layout = grid(2,1),
    link = :x,
    size = (1000,400))

#savefig("../../Master_2.0/figurar/RawData/pCO2/plot_pCO2_overview_B_C.pdf")
```

```
UndefVarError: plot_Bereiter not defined

Stacktrace:
 [1] top-level scope at In[187]:1
```

5 - Dust records

- Dust concentration record from EDC ice core (*Lambert et al., 2008*), spanning the last 800 kyr.
- Record of Fe mass accumulation rate in the Southern Ocean (*Martinez-Garcia et al.,2011)*, spanning the last 4.3 Myr.

5.1 - Lambert dust record

- Ice core record of dust (flux?) spanning the last 800 kyrs, from Epica Dome C (East Antarctic ice sheet).
- From *Lambert et al. (2008)*, denoted **L**.
- Dust concentration record from EDC ice core (East Antarctica), spanning the last 800 kyr
- data available from Pangaea, DOI: https://doi.org/10.1594/PANGAEA.695995 (https://doi.org/10.1594/PANGAEA.695995) .
- *Note: we are using the lpc-data (laser scatter method), since this has higher resolution downcore.*

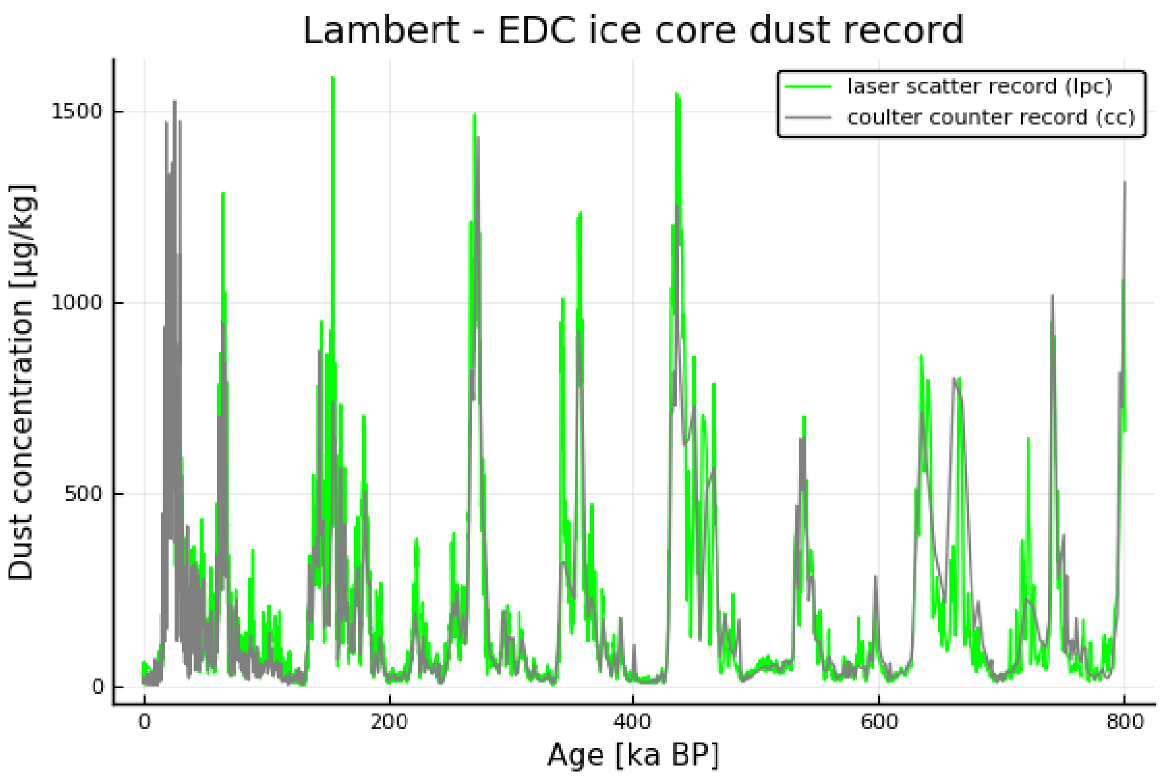
i) Load in data

Loading in data from Lambert, and checking which dataset is better to use - cc (coulter counter) or lpc (laser scatter)?

In [179]:

```
# Loading in data from Lambert
# and checking which dataset is better to use - cc (coulter counter) or lpc (las
er scatter)?


# load in Lambertdataset 1: EDC_dust_cc.tab
# From Pangaea data repository: Lambert et al (2008), file name EDC_dust_cc.tab
EDC_dust_cc = "../../MASTER_2.0/data/dust/Lambert(2008)_dust_EDC/datasets/EDC_du
st_cc.tab"
rawD_L_cc = readdlm(EDC_dust_cc, '\t', Float64, dims = (1154, 3), skipstart = 16
) #reading in the raw dataset, skipping the first 16 rows (descriptive)

# renaming columns of interest
depth_L_cc = rawD_L_cc[:,1] # depth (m)
age_L_cc = rawD_L_cc[:,2];  # age (ka) from Lambert dataset (EDC3 age model)
dust_L_cc = rawD_L_cc[:,3]; # dust concentration(μg/kg)




##### load in Lambert dataset 2: EDC_dust_lpc
# From Pangaea data repository: Lambert et al (2008), file name EDC_dust_lpc.tab
EDC_dust_lpc = "../../MASTER_2.0/data/dust/Lambert(2008)_dust_EDC/datasets/EDC_d
ust_lpc.tab"
rawD_L_lpc = readdlm(EDC_dust_lpc, '\t', Float64, dims = (5163, 3), skipstart =
15) #reading in the raw dataset, skipping the first 16 rows (descriptive)

# renaming columns of interest
depth_L_lpc = rawD_L_lpc[:,1]  # depth (m)
age_L_lpc   = rawD_L_lpc[:,2]; # age (ka) (from EDC3 age model)
dust_L_lpc  = rawD_L_lpc[:,3]; # dust concentration (μg/kg)



#################################
# Which data to use, lpc or cc?

length(dust_L_cc)  # 1154
length(dust_L_lpc) # 5163

# lpc has more observations
# looks like cc is concentrated in the upper end of the core
# based on visuals, seems that lpc has better resolution down-core

plot(xlabel = "Age [ka BP]", ylabel = "Dust concentration [μg/kg]", title = "Lam
bert - EDC ice core dust record")
plot!(age_L_lpc, dust_L_lpc, color = :lime,  label = "laser scatter record (lp
c)")
plot!(age_L_cc, dust_L_cc, color = :grey, label = "coulter counter record (cc)")

###### Note: #####################
# We decide to use the lpc-data (laser sensor),
# since this is the dataset with highest resolution back in time.
```

Out[179]:


Lambert - EDC ice core dust record
laser scatter record (lpc)
coulter counter record (cc)
Dust concentration [µg/kg]
0
500
1000
1500
0
200
400
600
800
Age [ka BP]

Dust concentration vs dust flux.

"Because of the low accumulation rate at Dome C (ca 3cm/yr water equivalent), dry deposition is dominant and **the atmospheric dust load is best represented by the dust flux**" (Lambert et al. (2008), referring to the work of Wolff et al (2006)). "(...)the flux is the better analogue for atmospheric concentration at Dome C, although its use could induce an error in the change in concentration between glacial and interglacial of 10-20%." (Wolff et al 2006, in supplementary discussion)

Unfortunately, Lambert et al. has **only published data as dust concentration, and not included it's conversion to dust flux**. They cite Wolff et al (2006). for how they have done the conversion from concentration to flux. Wolff et al (2006) report the following on how to to calculate the flux:

"flux, J, is given by:

$$J = vd \bullet C_{air} + K \bullet P \bullet C_{air},$$

where C_{air} is the concentration of the chemical in air, P is the snow precipitation rate, vd and K are constants of proportionality (vd is known as the dry deposition velocity, and K is related to a mass-based scavenging ratio). The first term is the dry deposited flux and the second term is the wet deposited flux.

The concentration in ice, C_{ice} is simply the flux divided by the average precipitation rate for the period represented by the ice sample."

Since we do not have access to the parameters Lambert et al. have used in the conversion (most importantly the precipitation rate), we check by comparing the Lambert dust concentration record to the Wolff Fe flux record.

> Or, Lambert et al report a mean precipitation rate for Dome C to be ca 3 cm/yr. However, reduced to a constant, this will not make any change to the dynamics of the time series, will it? All it would change is we could add the 10-20 \% uncertainty proposed [but was only proposed between glacial and interglacial] (supplementary discussion by Wolff et al.,2006)

In [180]:

```
fn = "/Users/maria/Downloads/41586_2006_BFnature04614_MOESM3_ESM_missingscut.tx
t"
data = readdlm(fn, '\t', dims = (272,5), Any, '\r', skipstart = 6)
Age = data[:,1]
Feflux = data[:,3]
Feflux[Feflux .== ""] .= NaN
```

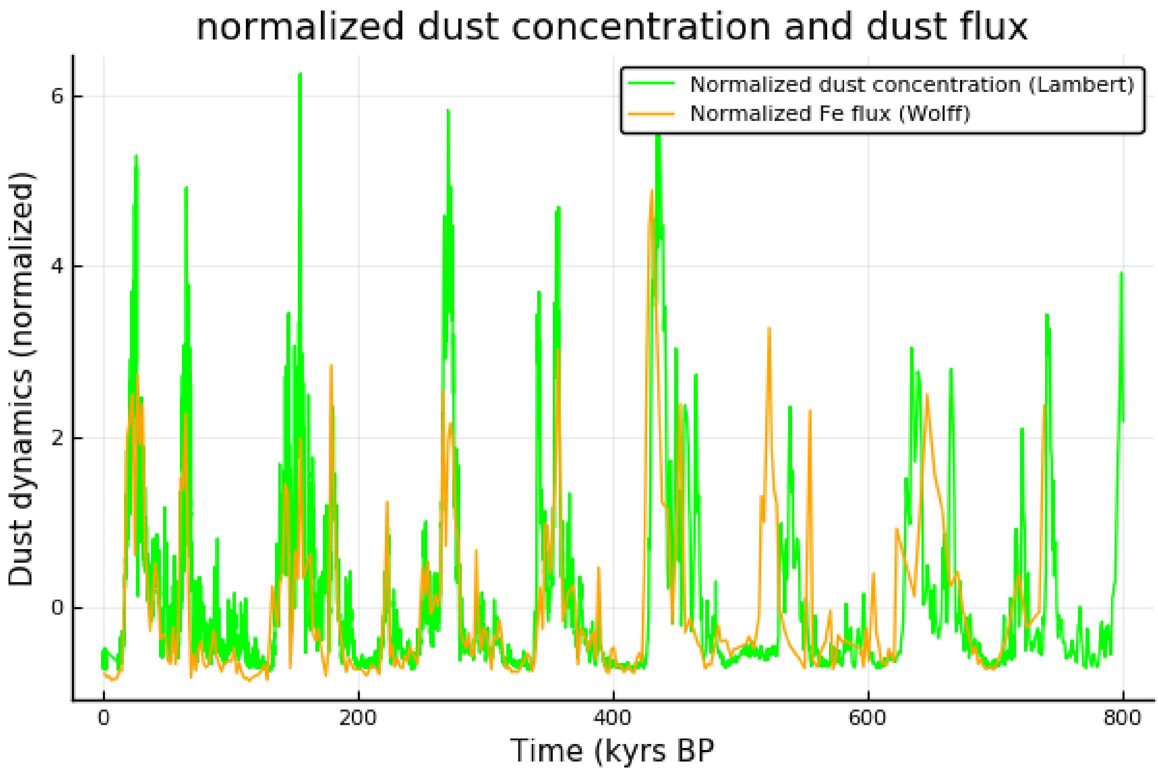
Out[180]:

```
0-element view(::Array{Any,1}, Int64[]) with eltype Any
```

In [181]:

```
# normalization to get comparable scale for data
norm_Feflux    = ((Feflux .- mean(Feflux)) / std(Feflux))
norm_dust_L    = ((dust_L_lpc .- mean(dust_L_lpc)) / std(dust_L_lpc))



# Plot concentration and flux to compare
plot(title = "normalized dust concentration and dust flux", xlabel = "Time (kyrs
BP", ylabel = "Dust dynamics (normalized)")
plot!(age_L_lpc, norm_dust_L, color = :lime, label = "Normalized dust concentrat
ion (Lambert)")
plot!(Age, norm_Feflux, color = :orange, label = "Normalized Fe flux (Wolff)")
```

Out[181]:



Lambert dust concentration data seems more "spikey" than the Wolff Fe flux data.

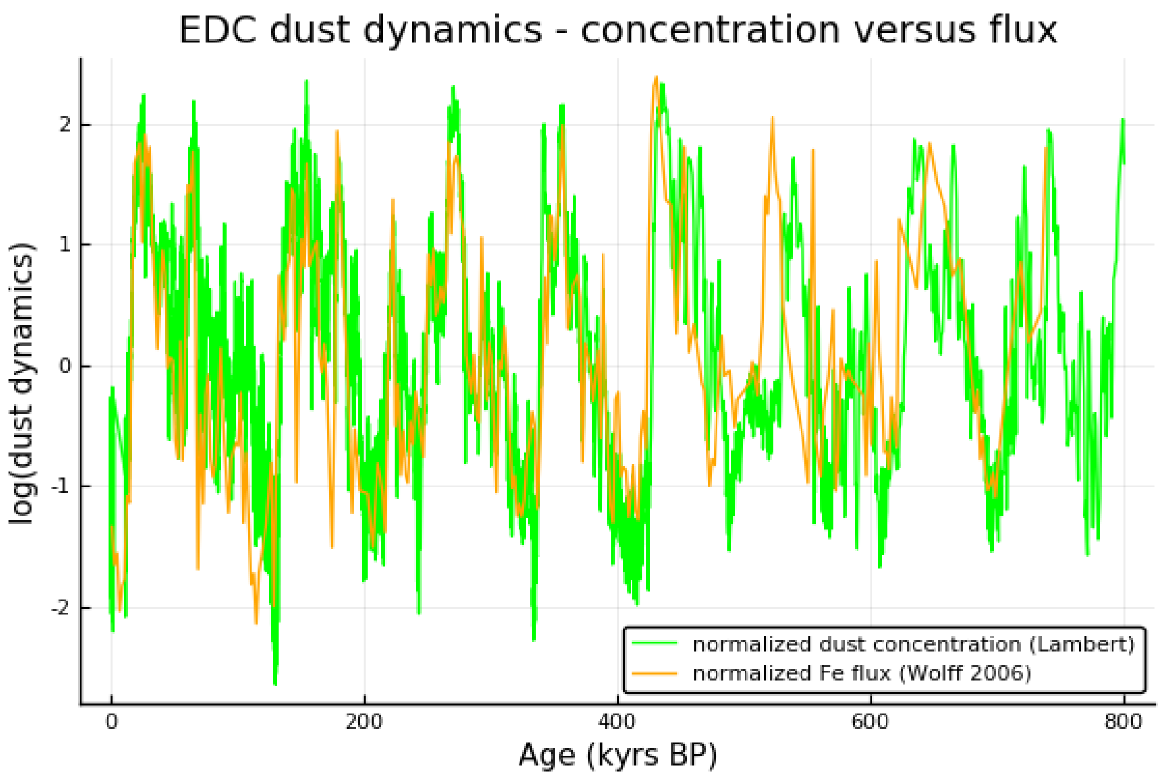
Set to a logarithmic scale

> (why? if the data was normalized already, this was a dynamics thing, not relict of units)

In [182]:

```
# NB can only run this cell once

# normalize Fe flux & put on a logarithmic scale
index = Feflux .!== NaN
norm_Fe = Feflux
norm_Fe[index] = log.(norm_Fe[index]) # log causes a -Inf
ix = (norm_Fe .!== -Inf) .& (norm_Fe .!== NaN)
norm_Fe[ix] = (norm_Fe[ix] .- mean(norm_Fe[ix])) ./ std(norm_Fe[ix])

# normalize dust concentration & put on a logarithmic scale
index = dust_L_lpc .!== NaN
norm_dustconc = dust_L_lpc
norm_dustconc[index] = log.(norm_dustconc[index]) # log causes a -Inf
ix = (norm_dustconc .!== -Inf) .& (norm_dustconc .!== NaN)
norm_dustconc[ix] = (norm_dustconc[ix] .- mean(norm_dustconc[ix])) ./ std(norm_d
ustconc[ix])


plot(xlabel = "Age (kyrs BP)", ylabel = "log(dust dynamics)", title = "EDC dust
 dynamics - concentration versus flux")
plot!(age_L_lpc, norm_dustconc, label = "normalized dust concentration (Lamber
t)", color = :lime)
plot!(Age, norm_Fe, label = "normalized Fe flux (Wolff 2006)", color = :orange )
```

Out[182]:



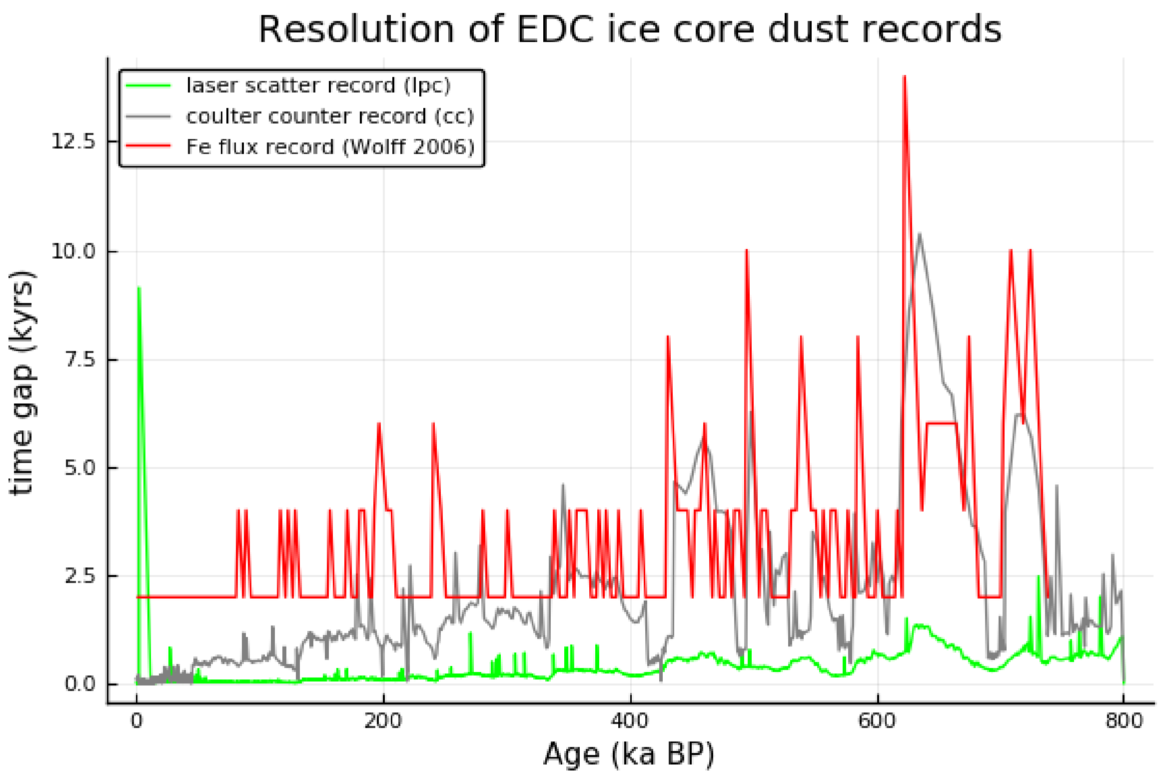
Dust concentration and flux are in fairly good agreement. There seem to be some differences in the age model, but this may be due to the Lambert record being on the EDC3 age model, while the Wolff data is on the EPC2 age model.

Check record resolutions

In [166]:

```
plot(xlabel = "Age (ka BP)", ylabel = "time gap (kyrs)", title = "Resolution of EDC ice core dust records")
plot!(age_L_lpc, diff(age_L_lpc), color = :lime,  label = "laser scatter record (lpc)")
plot!(age_L_cc, diff(age_L_cc), color = :grey, label = "coulter counter record (cc)")
plot!(Age, diff(Age), color = :red, label = "Fe flux record (Wolff 2006)") # low resolution, not suitable for our method.
```

Out[166]:



The Wolff (2006) Fe flux data has an order of magnitude lower resolution, which is what is of greater importance to our method (dynamical information). We therefore, in spite of not being as much of a direct process proxy, choose to use the Lambert dust concentration record further in our analyses.

We decide to use the lpc-data over the cc-data, since this is the dataset with highest resolution back in time.

Cutting away the low resolution interval at the youngest part of the record

There is a large time gap in the most recent part of the record, shown in the plot above. But where exactly is the gap? If we run the cell below (commented out for visual reasons) we can see that the observational gap is between 2.424 and 11.554 kyrs BP.

In [195]:

```
# show(age_L_lpc)
```

Now, we would like to cut the time series to only keep the high resolution part of the record.

In order to cut the time series, we need to find the corresponding index of where we want to cut. Therefore, we make an array of the observations up until that age gap point.

In [196]:

```
beforetimegap = [0.371, 0.383, 0.442, 0.454, 0.466, 0.478, 0.49, 0.503, 0.516,
0.529, 0.542, 0.555, 0.568, 0.581, 0.594, 0.606, 0.619, 0.632, 0.672, 0.701, 0.7
15, 0.729, 0.744, 0.759, 0.772, 0.786, 0.799, 0.825, 0.837, 0.862, 0.875, 0.889,
0.903, 0.917, 0.932, 0.947, 0.961, 0.976, 0.99, 1.005, 1.019, 1.032, 1.047, 1.06
1, 1.075, 1.089, 1.104, 1.118, 1.133, 1.147, 1.161, 1.174, 1.188, 1.201, 1.215,
1.229, 1.244, 1.258, 1.273, 1.287, 1.302, 1.316, 1.33, 1.345, 1.359, 1.373, 1.38
7, 1.401, 1.416, 1.432, 1.448, 1.463, 1.479, 1.494, 1.51, 1.525, 1.541, 1.556,
1.572, 1.588, 1.604, 1.621, 1.637, 1.654, 1.67, 1.686, 1.702, 1.737, 1.754, 1.77
1, 1.787, 1.804, 1.822, 1.839, 1.856, 1.873, 1.889, 1.905, 1.922, 1.938, 1.956,
1.973, 1.99, 2.007, 2.025, 2.044, 2.061, 2.078, 2.095, 2.113, 2.13, 2.147, 2.165
, 2.184, 2.202, 2.219, 2.237, 2.253, 2.271, 2.289, 2.306, 2.327, 2.347, 2.367,
2.405, 2.424] # copied from running the cell above.
length(beforetimegap) # 126
# checking the age value of the index before and after time gap
age_L_lpc[126] # 2.424
age_L_lpc[127] # 11.554 # high resolution array starts at index 127

# From the plot above, the time gap should be around 9 kyrs long.
age_L_lpc[127] - age_L_lpc[126]
# So this looks right
```

Out[196]:

```
9.13
```

We have found the large gap in resolution, it's between index 126 and 127. We therefore cut the first 126 observations out of the time series.

In [197]:

```
# cutting the first 126 elements of the arrays, and rename the new, shorter arra
ys

# age array
print("The original array is ", length(age_L_lpc), " elements long") # long vers
ion is 5163 elements
age_L_EDC3 = age_L_lpc[(age_L_lpc .> age_L_lpc[126])]               # age (ka)
 (from EDC3 age model)
print("
The cut array is ", length(age_L_EDC3), " elements long")

#  depth array
depth_L = depth_L_lpc[(age_L_lpc .> age_L_lpc[126])];
# checking that we cut right
depth_L_lpc[126]  # 99 m
depth_L_lpc[127] # 358 m
# large gap, all good

# dust array
dust_L = dust_L_lpc[(age_L_lpc .> age_L_lpc[126])]
# checking that we cut right
length(dust_L_lpc) # 5163
length(dust_L) # 5037
#same as above, all good
```

```
The original array is 5163 elements long
The cut array is 5037 elements long
```

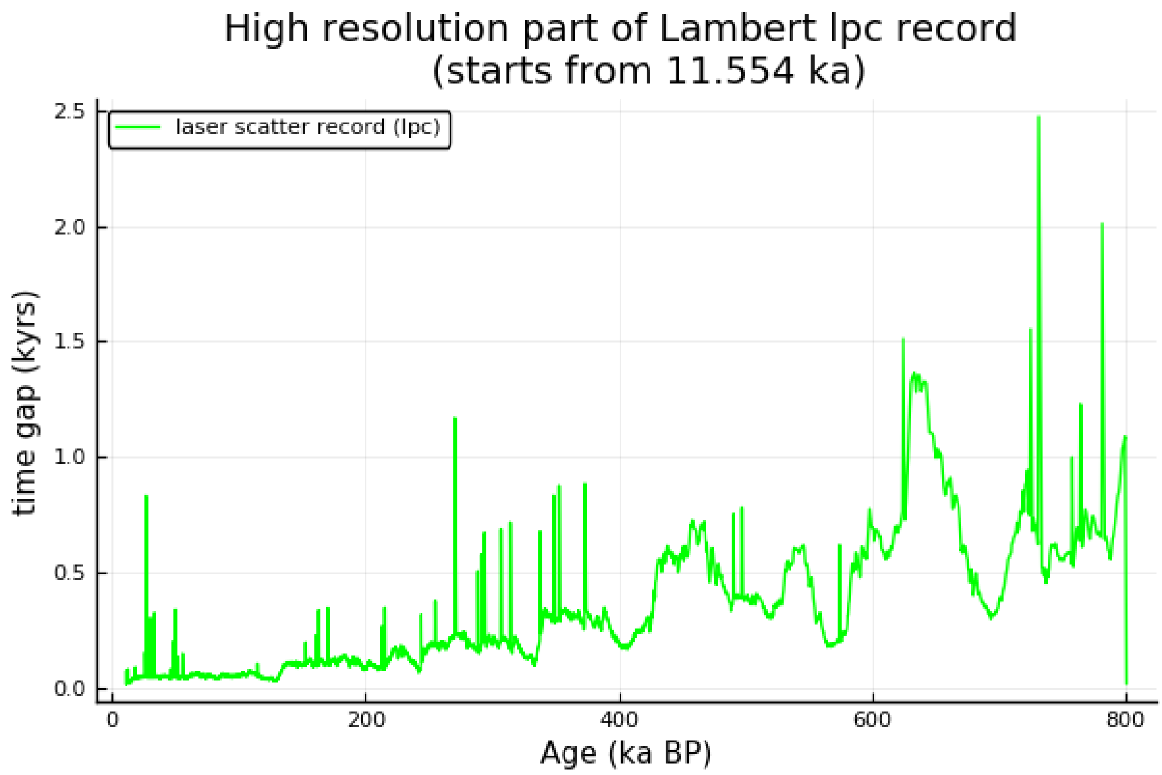
Out[197]:

```
5037
```

In [198]:

```
plot(xlabel = "Age (ka BP)", ylabel = "time gap (kyrs)", title = "High resolutio
n part of Lambert lpc record
    (starts from 11.554 ka)")
plot!(age_L_EDC3, diff(age_L_EDC3), color = :lime,  label = "laser scatter recor
d (lpc)")
```

Out[198]:



Adjustments to age model

Note: We have two datasets (dust and CO2) from the same ice core (Epica Dome C). However, different age models are used on the two datasets. The dust record from Lambert et al. (2008) uses the older EPC3 age model, while the CO2 record from Bereiter et al. (2015) uses the revised AICC2012 age model. In the following, we want to get both datasets on the same age model (AICC2012).

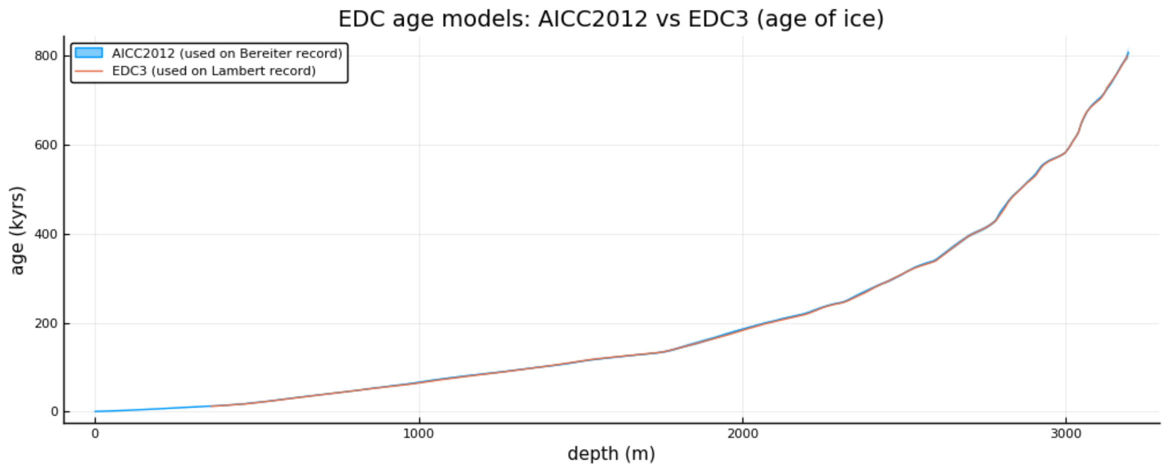
In [199]:

```
# read in data for the AICC2012 age model

filepath_AICC2012 = "../../MASTER_2.0/Koding/age_models/EDC_AICC2012_chron.tab"
rawD_AICC2012 = readdlm(filepath_AICC2012, skipstart = 23, dims = (1936,8))

#columns
depth      = Array{Float64,1}(rawD_AICC2012[:,1])   # Depth ice/snow [m]
age_ice    = Array{Float64,1}(rawD_AICC2012[:,2])   # Age [ka BP]
age_ice_1σ = Array{Float64,1}(rawD_AICC2012[:,3])   # Age std dev [±]
age_gas    = Array{Float64,1}(rawD_AICC2012[:,4])   # Gas age [ka BP]
age_gas_1σ = Array{Float64,1}(rawD_AICC2012[:,5])   # Gas age std dev [±]
# acc_rate = rawD_AICC2012[:,6]    # Acc rate ice per year [m/a]
# tf = rawD_AICC2012[:,7]          # Thinning functionTF
# lidie      = Array{Float64,1}(rawD_AICC2012[:,8])      # Lock-in depth in ice
 equivalent (LIDIE) [m] #
;
```

In [200]:

```
# Show difference between AICC2012 and EDC3 age models

plot_agemodels =
plot(title = "EDC age models: AICC2012 vs EDC3 (age of ice)",
    size = (1000, 400),
#     xlims = (2500,2510),
    xlabel = "depth (m)",
    ylabel = "age (kyrs)")
!(depth, age_ice,
    label = "AICC2012 (used on Bereiter record)",
    ribbon = (age_ice_1σ, age_ice_1σ),
    fillalpha = 0.5)
plot!(depth_L, age_L_EDC3,
    label = "EDC3 (used on Lambert record)")
#plot!(depth, lidie)
```

Out[200]:



In [ ]:

```

```

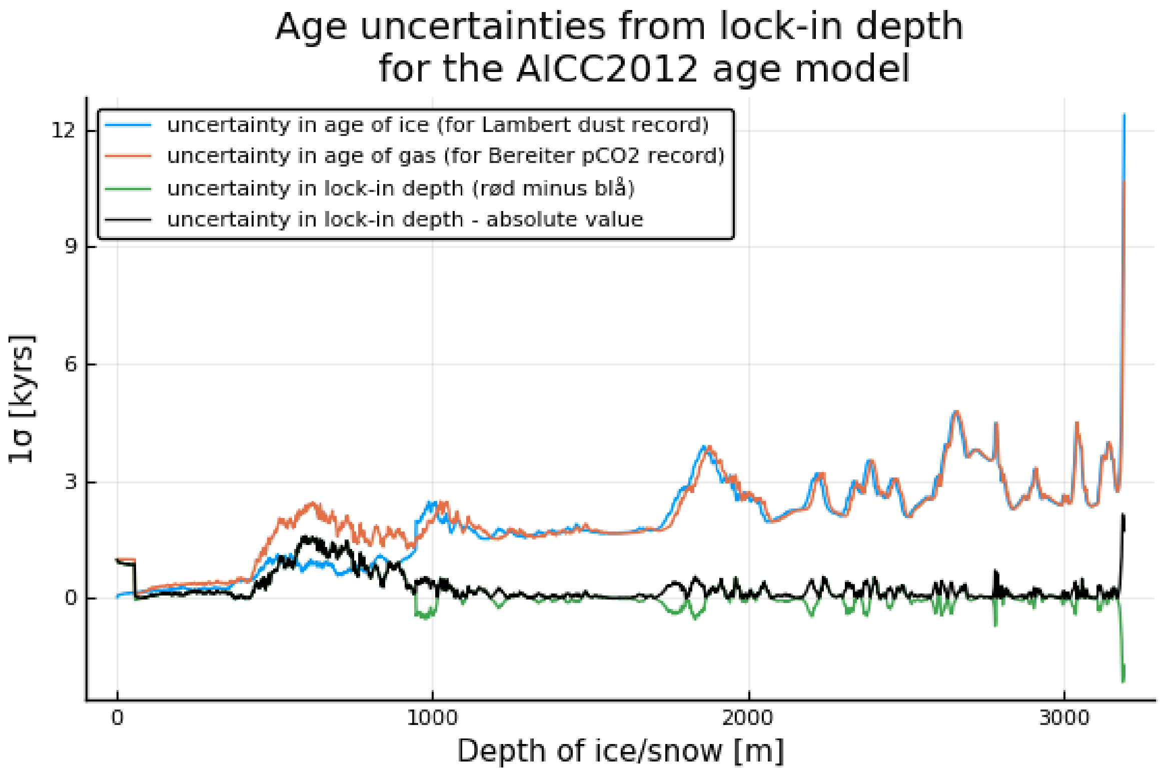
The differences between the two age models are very subtle, barely noticeable in the plot above. However, we make a point out of having the Lambert and Bereiter records on the same age model, to minimize/eliminate the age uncertainty between these records. Another advantage of the newer AICC2012 age model is that it includes a quantification of the age uncertainty. The quality of age models are of particular importance for our approach, as we operate with a lead-lag-based definition of causality.

Age uncertainty from lock-in depth of gas in ice: For analyses between Lambert and Bereiter, both from the EDC core, we shall only use the age uncertainty from lock-in time of the gas bubbles in the ice. This is not reported directly in the dataset with the age model, but we can derive an array representing this uncertainty.

In [214]:

```
# age uncertainty between age of gas and age of ice

edc_gasice_ageunc_1σ = age_gas_1σ - age_ice_1σ
# for some reason,  which uncertainty is larger varies. we therefore take the ab
solute value

edc_gasice_ageunc_1σ_absolutevalue = [abs(i) for i in edc_gasice_ageunc_1σ]

#plot age uncertainties
plot(xlabel = "Depth of ice/snow [m]",
    ylabel = "1σ [kyrs]",
    title = "Age uncertainties from lock-in depth
    for the AICC2012 age model")
plot!(depth, age_ice_1σ, label = "uncertainty in age of ice (for Lambert dust re
cord)")
plot!(depth, age_gas_1σ, label = "uncertainty in age of gas (for Bereiter pCO2 r
ecord)")
plot!(depth, edc_gasice_ageunc_1σ, label = "uncertainty in lock-in depth (rød mi
nus blå)", ms = :dot)
plot!(depth, edc_gasice_ageunc_1σ_absolutevalue, label = "uncertainty in lock-in
depth - absolute value", c = :black)
```

Out[214]:

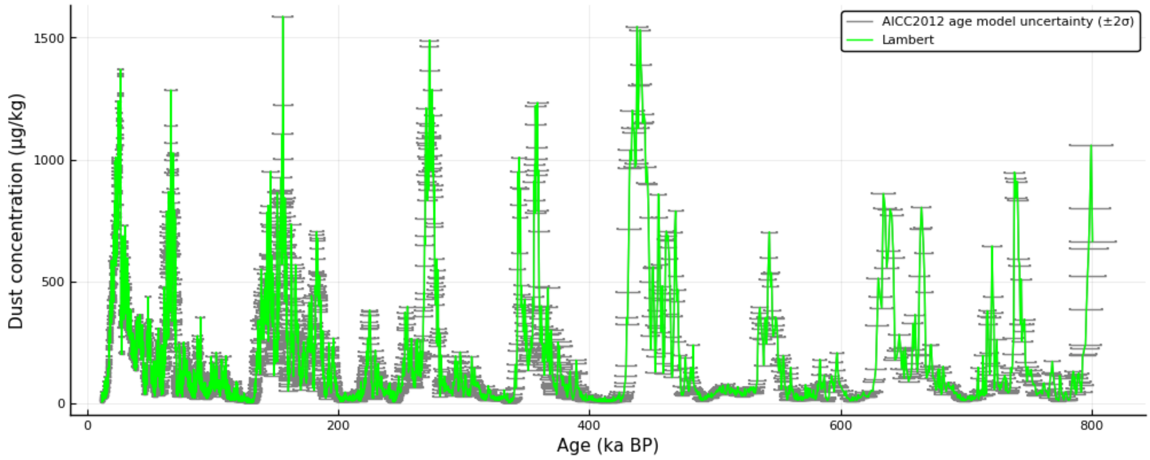


In [217]:

```
# Interpolation of data to the AICC2012 age model

# (reformulates from array of discrete ages for discrete depths, to ages given a
s a continuous function of any depth)
interpolate_ageice    = LinearInterpolation(age_ice, depth)     # age of ice as
 a function of depth
interpolate_ageice_1σ = LinearInterpolation(age_ice_1σ, depth)  # with associate
d uncertainties (1σ)
interpolate_agegas    = LinearInterpolation(age_gas, depth)     # age of gas as
 a function of depth
interpolate_agegas_1σ = LinearInterpolation(age_gas_1σ, depth)  # with associate
d uncertainties (1σ)
interpolate_icegas_ageunc_1σ = LinearInterpolation(edc_gasice_ageunc_1σ_absolute
value, depth) # age uncertainty between gas and ice (1σ)
;

# Give corresponding age values from interp_AICC2012 to the depth of observation
s in the Lambert dataset
newages_L     = [interpolate_ageice(i) for i in depth_L]    # array of ages (ka)
 for the Lambert dust data, according to the AICC2012 age model for ice.
newages_1σ_L = [interpolate_ageice_1σ(i) for i in depth_L] # associated age unce
rtainties
newages_1σ_L_edc = [interpolate_icegas_ageunc_1σ(i) for i in depth_L] # 1σ age un
certainty between gas and ice - to be used for analysis between Bereiter and Lam
bert records (both from edc)

;
```

In [218]:

```
# plot

#plot_Lambert_newagemodel =
plot(#title = "EDC dust concentration on AICC2012 age model",
    size = (1000, 400),
    xlabel = "Age (ka BP)",
    ylabel = "Dust concentration (μg/kg)")
plot!(newages_L, dust_L,
    xerr = 2 * newages_1σ_L, # 2σ = 95% confidence interval on age
    ms = 1,
    color = :grey,
    label = "AICC2012 age model uncertainty (±2σ)")
plot!(newages_L, dust_L,
    color = :lime,
    label = "Lambert")

#savefig("../figurar/RawData/Dust/Lambert_aicc2012_wAgeUnc.pdf")
```

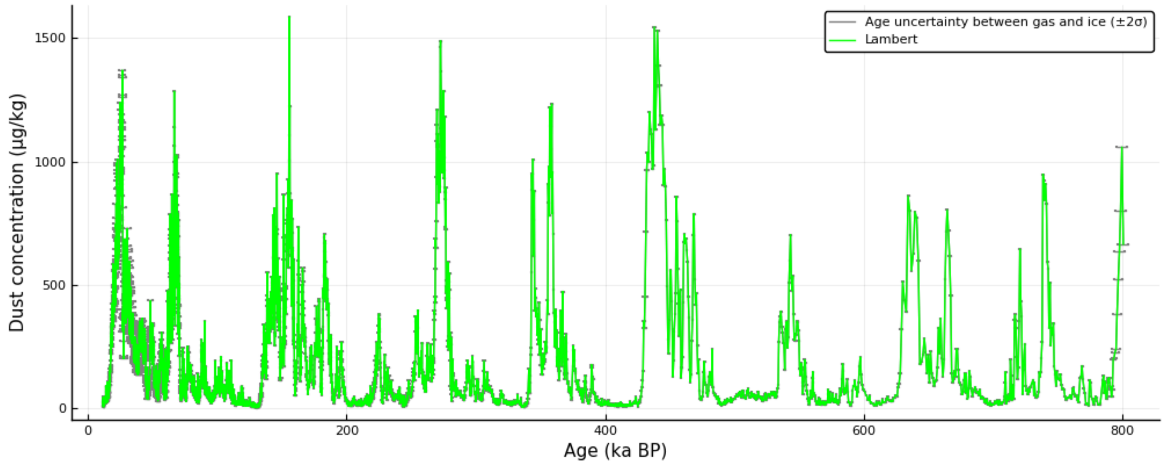


We note that the age uncertainty is quite large in this record. What effect may this have on the high resoultion analysis?

In [219]:

```
# plot

#plot_Lambert_newagemodel =
plot(#title = "EDC dust concentration on AICC2012 age model",
     size = (1000, 400),
     xlabel = "Age (ka BP)",
     ylabel = "Dust concentration (μg/kg)")
plot!(newages_L, dust_L,
     xerr = 2 * newages_1σ_L_edc, # ±2σ = 95% confidence interval
     ms = 1,
     color = :grey,
     label = "Age uncertainty between gas and ice (±2σ)")
plot!(newages_L, dust_L,
     color = :lime,
     label = "Lambert")

savefig("../figurar/RawData/Dust/Lambert_aicc2012_wAgeUnc_between_gasice.pdf")
```

Out[219]:



ii) Reverse dataset

Redefine from age (increasing backwards) to time (increasing forwards)

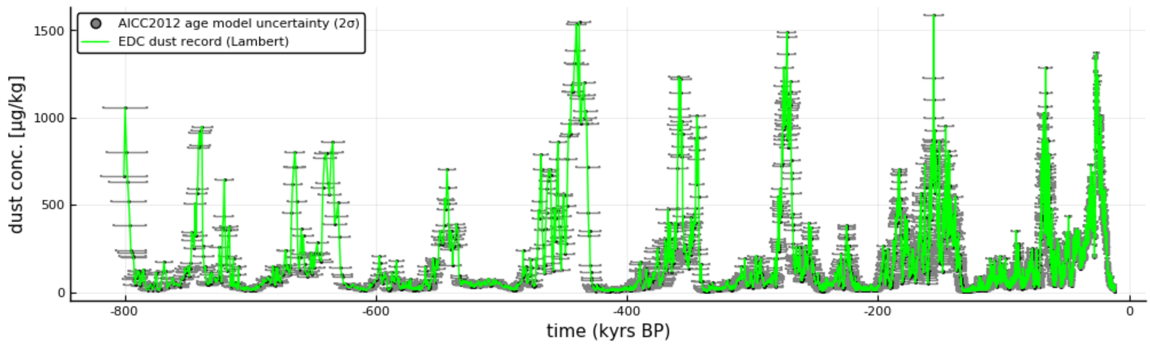
In [221]:

```
# redefine Lambert dataset from age (increasing backwards with indexation) to ti
me (increasing forwards with indexation)
t_L     = - reverse(newages_L) # time is negative because we are defining present
as 0
t_1σ_L = reverse(newages_1σ_L)
t_1σ_L_edc = reverse(newages_1σ_L_edc)
dust_L = reverse(dust_L)
;
@save "../Koding/WrangledDataFiles/BasicArrays/Lambert.jld2" t_L t_1σ_L t_1σ_L_e
dc dust_L
```

> DERIVERE FOR Å FÅ **FLUX**? HJELP

In [131]:

```
# plot Lambert time series
@load "../Koding/WrangledDataFiles/BasicArrays/Lambert.jld2"

plot_Lambert_time =
plot(#title = "EDC aeolian dust record (Lambert lpc record)",
    size = (1000, 300),
    xlabel = "time (kyrs BP)",
    ylabel = "dust conc. [μg/kg]")
scatter!(t_L, dust_L,
    xerr = 2 * t_1σ_L,  # AICC2012 age uncertainty (2σ)
    ms = 1,
    color = :grey,
    label  = "AICC2012 age model uncertainty (2σ)"
    )
plot!(t_L, dust_L,
    color = :lime,
    label = "EDC dust record (Lambert)"
    #label = "Lambert"
    )

savefig("../figurar/RawData/Dust/Lambert_aicc2012_wTimeUnc.pdf")
```

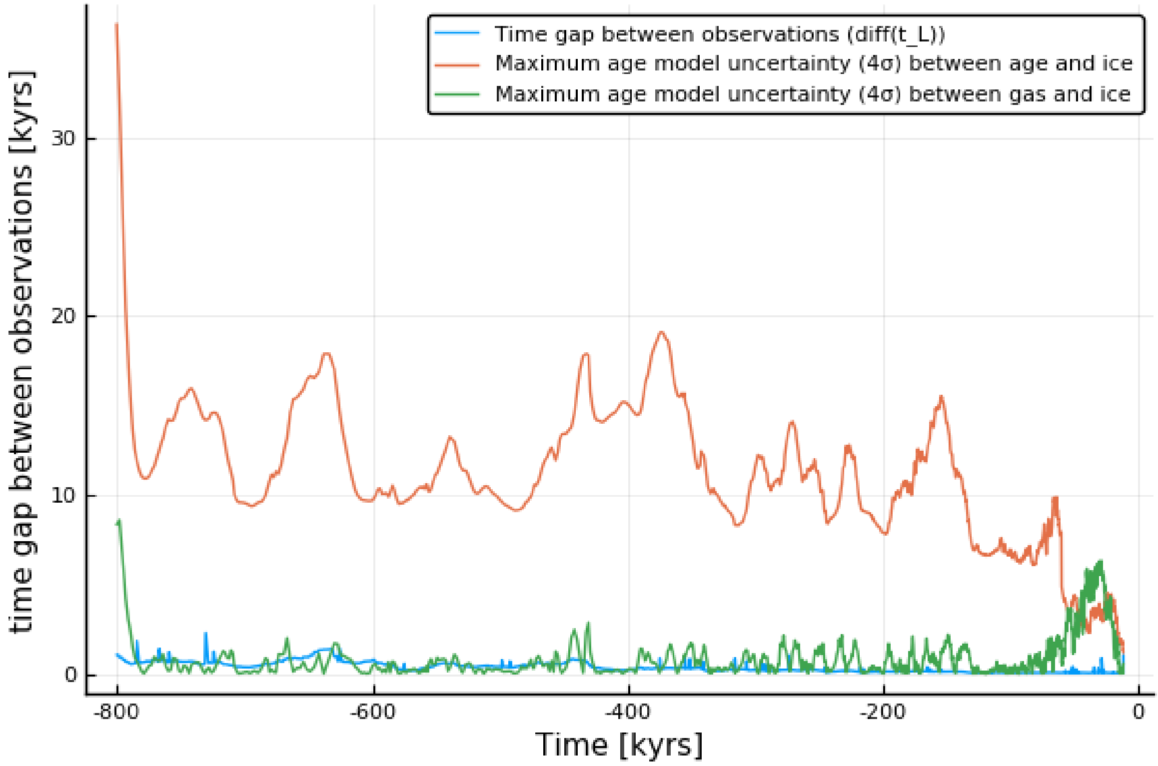


iv) Interpolation of data, to ensure we have data in all bins.

In [143]:

```
# Need for interpolation of Lambert data?

plot(t_L, diff(t_L),
    xlabel = "Time [kyrs]",
    ylabel = "time gap between observations [kyrs]",
    label = "Time gap between observations (diff(t_L))")
plot!(t_L, 4*t_1σ_L, label = "Maximum age model uncertainty (4σ) between age and ice")
plot!(t_L, 4*t_1σ_L_edc, label = "Maximum age model uncertainty (4σ) between gas and ice")
```

Out[143]:



In [136]:

```
maximum(diff(t_L)) # 2.3 kyrs  # (9.3 kyrs if not cutting the youngest part of t
he record)
minimum(diff(t_L)) # 20 years
mean(diff(t_L))    # 157 years # (155 years if not cutting the youngest part of
 the record)

# Overall high resolution, but yes, we need interpolation for some bins
```

Out[136]:

```
0.1566320492454329
```

Resolution of the Lambert data is on average one datapoint every 150-200 years, but there are gaps of over 2000 years between observations. This means we need to interpolate.

> **No need to interpolate, as long as age uncertainty is larger than the time gaps - values will then still be drawn for each bin through resampling.** We might still need to interpolate for the version that is to be run analysis with EDC pCO2 (reduced age uncertainty). Let's check.

In [142]:

```
##### redefining the data as an UncertainIndexValueDataset

t_uiv_L = [UncertainValue(Normal, t_L[i], t_1σ_L[i]) for i in 1:length(t_L)] # a
ge uncertainty between gas and ice
dust_uiv_L = [UncertainValue(Normal, dust_L[i], 0) for i in 1:length(dust_L)]
uivD_L_noIntp = UncertainIndexValueDataset(t_uiv_L, dust_uiv_L)

# uivD_L_EDC is to be analysed with another EDC core, and therefore we don't inc
lude the age model uncertainties
t_uiv_L_EDC = [UncertainValue(Normal, t_L[i], t_1σ_L_edc[i]) for i in 1:length(t
_L)] # age uncertainty between gas and ice
dust_uiv_L = [UncertainValue(Normal, dust_L[i], 0) for i in 1:length(dust_L)]
uivD_L_EDC_noIntp = UncertainIndexValueDataset(t_uiv_L_EDC, dust_uiv_L)

##### Defining the grid for binned resampling
binsize = 1 # each timestep is 1000 years
tmin_L = ceil(minimum(t_L))   # first bin midpoint
tmax_L = floor(maximum(t_L))  # last bin midpoint
grid_L = tmin_L - binsize/2 : binsize : tmax_L + binsize/2 # these must be the b
in edges if the bin midpoints shall be at every whole 1 kyr


# resampling on the grid
resampling_method_L = BinnedResampling(grid_L, 1000) # resample 1000 draws (with
substitution) in each bin of the grid
@time L_binned_full_noIntp = resample(uivD_L_noIntp, resampling_method_L)
#@time L_binned_full_noIntp_edc = resample(uivD_L_EDC_noIntp, resampling_method_
L)
```

```
 70.109201 seconds (58.62 M allocations: 19.675 GiB, 15.16% gc time)
```

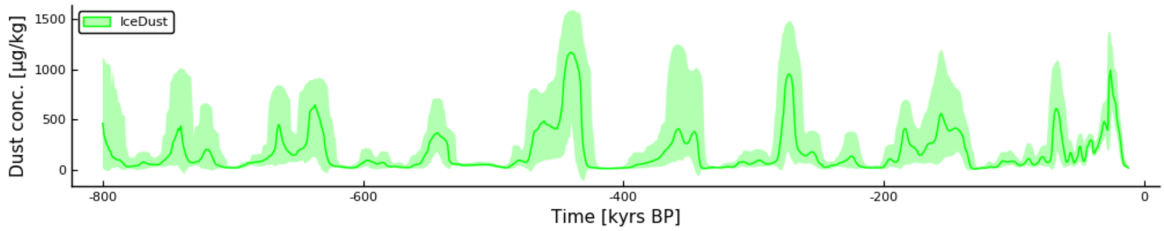
Out[142]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 789 uncertain values coupled with 789 uncertain indic
es
```

In [152]:

```
######### plot the binned resampled Lambert dust time series with reduced age un
certainty
ts = L_binned_full_noIntp
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_L_edc =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :lime,
    label =  "IceDust",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc. [μg/kg]",
    grid = false,
    size = (1000,200),
    legend = :topleft
    )
```

Out[152]:



- check if the binned resampling could draw a value in every bin when Lambert record was defined with the smaller gas-ice age uncertainty

In [149]:

```
show(L_binned_full_noIntp_edc)
```

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 789 uncertain values coupled with 789 uncertain indic
es
```

In [150]:

```
######### plot the binned resampled Lambert dust time series with reduced age un
certainty
ts = L_binned_full_noIntp_edc
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_L_edc =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :olive,
    label =  "IceDust - age uncertainty between gas and ice",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc. [μg/kg]",
    grid = false,
    size = (1000,200),
    legend = :left
    )
# ...contains empty cells..., we need to interpolate..
```

```
ArgumentError: invalid index: nothing of type Nothing

Stacktrace:
 [1] to_index(::Nothing) at ./indices.jl:270
 [2] to_index(::StepRangeLen{Float64,Base.TwicePrecision{Float64},Ba
se.TwicePrecision{Float64}}, ::Nothing) at ./indices.jl:247
 [3] to_indices at ./indices.jl:298 [inlined]
 [4] to_indices at ./indices.jl:295 [inlined]
 [5] getindex(::StepRangeLen{Float64,Base.TwicePrecision{Float64},Ba
se.TwicePrecision{Float64}}, ::Nothing) at ./abstractarray.jl:981
 [6] quantile(::UncertainScalarKDE{Float64}, ::Float64) at /Users/ma
ria/.julia/packages/UncertainData/Pb1tS/src/uncertain_values/Uncerta
inScalarsKDE.jl:113
 [7] _broadcast_getindex_evalf at ./broadcast.jl:625 [inlined]
 [8] _broadcast_getindex at ./broadcast.jl:598 [inlined]
 [9] getindex at ./broadcast.jl:558 [inlined]
 [10] copyto_nonleaf!(::Array{Float64,1}, ::Base.Broadcast.Broadcast
ed{Base.Broadcast.DefaultArrayStyle{1},Tuple{Base.OneTo{Int64}},type
of(quantile),Tuple{Base.Broadcast.Extruded{Array{Any,1},Tuple{Bool},
Tuple{Int64}},Float64}}, ::Base.OneTo{Int64}, ::Int64, ::Int64) at
./broadcast.jl:982
 [11] copy at ./broadcast.jl:836 [inlined]
 [12] materialize(::Base.Broadcast.Broadcasted{Base.Broadcast.Defaul
tArrayStyle{1},Nothing,typeof(quantile),Tuple{Array{Any,1},Float6
4}}) at ./broadcast.jl:798
 [13] top-level scope at In[150]:4
```

The edc version contains empty bins. Since we need continuous time series for our method, we need to interpolate some values if we are to use this record.

In [84]:

```
# Create continuous functions that linearly interpolates between each data point
s
interpolate_t_L     = LinearInterpolation(t_L, t_L)             # function to inte
rpolate age array
interpolate_t_1σ_L = LinearInterpolation(t_1σ_L, t_L)          # function to inte
rpolate age model uncertainties
interpolate_t_1σ_L_edc = LinearInterpolation(t_1σ_L_edc, t_L) # function to inte
rpolate age uncertainties between gas and ice
interpolate_dust_L = LinearInterpolation(dust_L, t_L);         # function to inte
rpolate dust concentration
```

The otherwise high resolution of the data (one observation every 150 years on average) makes it defendable to draw interpolated values on a high resolution/centennial resolution grid without creating many false values. This will allow us to run higher resolution analyses on this dataset.

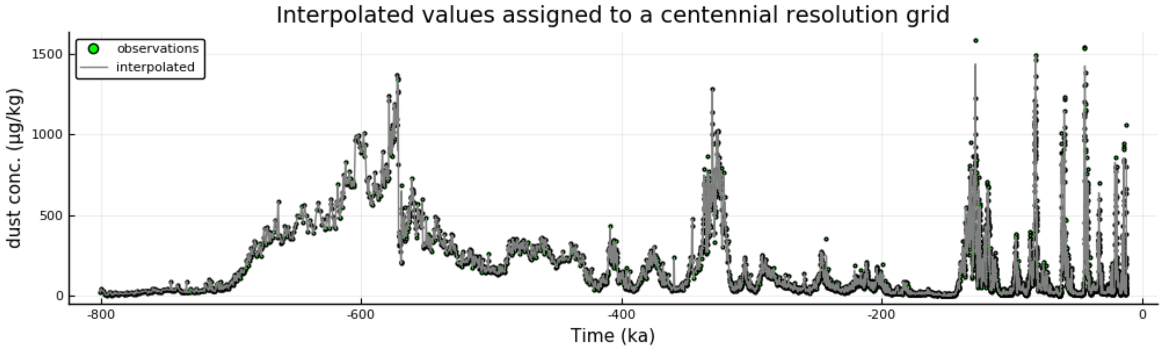
> WHAT RESOLUTION TO ASSIGN ON?

In [99]:

```
# make a fine-grained time-grid to which we assign interpolated values for the L
ambert dataset.
fine_grid_L = minimum(t_L) : 0.1 : maximum(t_L)  # 1 bin for every 100 years

# we then put the interpolated values in a new object intpD_L, containing one va
lue in every bin of the fine time grid
intpD_t_L = [interpolate_t_L(i) for i in fine_grid_L];
intpD_t_1σ_L = [interpolate_t_1σ_L(i) for i in fine_grid_L];
intpD_t_1σ_L_edc = [interpolate_t_1σ_L_edc(i) for i in fine_grid_L];
intpD_dust_L = [interpolate_dust_L(i) for i in fine_grid_L];

print(length(intpD_t_L))

# let's plot to check if interpolation is ok

plot(title = "Interpolated values assigned to a centennial resolution grid",
    xlabel = "Time (ka)",
    ylabel = "dust conc. (μg/kg)",
    size = (1000,300))
scatter!(t_L, dust_L,
    #xerr = 2 * t_1σ_L,
    ms = 2,
    color = :lime,
    label = "observations")
plot!(intpD_t_L, intpD_dust_L,
    #xerr = intpD_t_2σ_B,    # too computationally heavy and too cluttering to pl
ot xerr for every 100 years
    color = :grey,
    label = "interpolated")
```

```
7888
```

Out[99]:



Some observed peak values are not captured by the interpolation when we assign the values on a centennial grid.

To minimize the loss of nuance in data, we may choose a higher resolution fine grid for assigning the interpolated values.

> wouldn't that also require higher n_draws when moving on to BinnedResampling?

I think: finer grid = less loss of nuance, but also TRADE-OFF with creating a fake smooth data where we have larger time gaps.

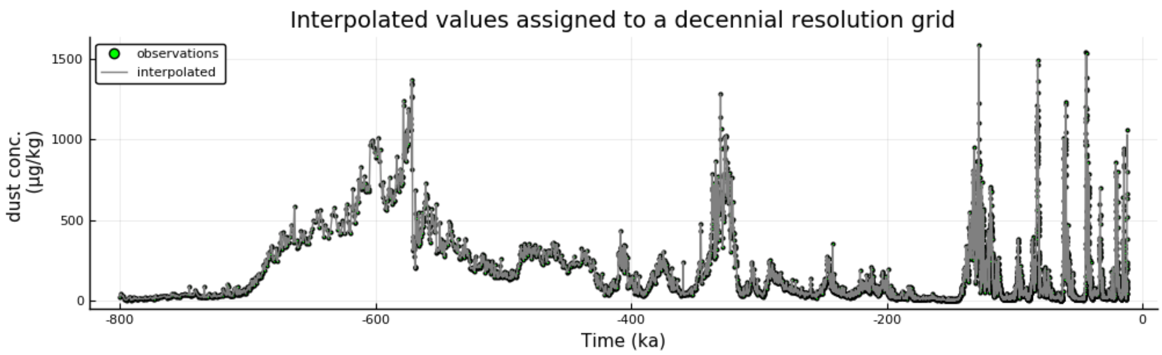
> DISCUSS with Bjarte and Jo the best way to go about this.

In [100]:

```
# make a fine-grained time-grid to contain interpolated values for the Lambert d
ataset: Decennial resolution
fine_grid_L = minimum(t_L) : 0.01 : maximum(t_L)  # 1 bin for every 10 years

# we then put the interpolated values in a new object intpD_L, containing one va
lue in every bin of the fine time grid
t_L_intpDdec = [interpolate_t_L(i) for i in fine_grid_L];
t_1σ_L_intpDdec = [interpolate_t_1σ_L(i) for i in fine_grid_L];
dust_L_intpDdec = [interpolate_dust_L(i) for i in fine_grid_L];

print(length(t_L_intpDdec))

# let's plot to check if interpolation is ok

plot(title = "Interpolated values assigned to a decennial resolution grid",
    xlabel = "Time (ka)",
    ylabel = "dust conc.
    (µg/kg)",
    size = (1000,300))
scatter!(t_L, dust_L,
    #xerr = 2 * t_1σ_L,
    ms = 2,
    color = :lime,
    label = "observations")
plot!(t_L_intpDdec, dust_L_intpDdec,
    #xerr = intpD_t_2σ_B,   # too computationally heavy and too cluttering to pl
ot xerr for every 100 years
    ms = 0.1,
    color = :grey,
    label = "interpolated")
```

78880

Out[100]:



Redefining the time series as an `UncertainIndexValueDataset`

In [101]:

```
# redefining the interpolated dataset ``intpD_L``(centennial resolution) as an u
ivD

t_uiv_L = [UncertainValue(Normal, intpD_t_L[i], intpD_t_1σ_L[i]) for i in 1:leng
th(intpD_t_L)] # age uncertainty from AICC2012 age model
dust_uiv_L = [UncertainValue(Normal, intpD_dust_L[i], 0) for i in 1:length(intpD
_dust_L)]      # no uncertainties reported concerning measurements
uivD_L = UncertainIndexValueDataset(t_uiv_L, dust_uiv_L)

# We now have our uncertain index value Dataset for the dust record from Lambert
et al.(2008)
```

Out[101]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 7888 uncertain values coupled with 7888 uncertain ind
ices
```

Given that the Bereiter (pCO2) and Lambert (dust) records are both from the same ice core and now on the same age model, we can ignore the age model uncertainty between them and analyse directly on depth. We will use only the smaller uncertainties stemming from the lock-in time of the gas bubbles in the ice. We therefore also prepare a uivD version of the Lambert record with this small age uncertainty, for analysis between that specific time series pair.

In [104]:

```
# uivD_L_EDC is to be analysed with another EDC core, and therefore we don't inc
lude the age model uncertainties

t_uiv_L_EDC = [UncertainValue(Normal, intpD_t_L[i], intpD_t_1σ_L_edc[i]) for i i
n 1:length(intpD_t_L)] # age uncertainty between gas and ice
dust_uiv_L = [UncertainValue(Normal, intpD_dust_L[i], 0) for i in 1:length(intpD
_dust_L)]
uivD_L_EDC = UncertainIndexValueDataset(t_uiv_L_EDC, dust_uiv_L)
```

Out[104]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 7888 uncertain values coupled with 7888 uncertain ind
ices
```

In [105]:

```
plot(uivD_L_EDC, xlabel = "UncertainIndex (Time in kyrs BP)", ylabel = "Uncertai
nValue (dust conc. in µg/kg)")
```

```
InterruptException:

Stacktrace:
 [1] get_clims(::Plots.Subplot{Plots.PyPlotBackend}) at /Users/mari
a/.julia/packages/Plots/qZHsp/src/utils.jl:556
 [2] get_clims(::Plots.Subplot{Plots.PyPlotBackend}, ::Plots.Series)
at /Users/maria/.julia/packages/Plots/qZHsp/src/utils.jl:569
 [3] py_add_series(::Plots.Plot{Plots.PyPlotBackend}, ::Plots.Serie
s) at /Users/maria/.julia/packages/Plots/qZHsp/src/backends/pyplot.j
l:402
 [4] _before_layout_calcs(::Plots.Plot{Plots.PyPlotBackend}) at /Use
rs/maria/.julia/packages/Plots/qZHsp/src/backends/pyplot.jl:975
 [5] prepare_output(::Plots.Plot{Plots.PyPlotBackend}) at /Users/mar
ia/.julia/packages/Plots/qZHsp/src/plot.jl:254
 [6] show(::Base64.Base64EncodePipe, ::MIME{Symbol("image/png")}, ::
Plots.Plot{Plots.PyPlotBackend}) at /Users/maria/.julia/packages/Plo
ts/qZHsp/src/output.jl:198
 [7] #base64encode#3(::Nothing, ::typeof(Base64.base64encode), ::Fun
ction, ::MIME{Symbol("image/png")}, ::Vararg{Any,N} where N) at /Use
rs/sabae/buildbot/worker/package_macos64/build/usr/share/julia/stdli
b/v1.2/Base64/src/encode.jl:206
 [8] base64encode(::Function, ::MIME{Symbol("image/png")}, ::Vararg
{Any,N} where N) at /Users/sabae/buildbot/worker/package_macos64/bui
ld/usr/share/julia/stdlib/v1.2/Base64/src/encode.jl:203
 [9] _ijulia_display_dict(::Plots.Plot{Plots.PyPlotBackend}) at /Use
rs/maria/.julia/packages/Plots/qZHsp/src/ijulia.jl:50
 [10] display_dict(::Plots.Plot{Plots.PyPlotBackend}) at /Users/mari
a/.julia/packages/Plots/qZHsp/src/init.jl:83
 [11] #invokelatest#1 at ./essentials.jl:790 [inlined]
 [12] invokelatest at ./essentials.jl:789 [inlined]
 [13] execute_request(::ZMQ.Socket, ::IJulia.Msg) at /Users/maria/.j
ulia/packages/IJulia/F1GUo/src/execute_request.jl:112
 [14] #invokelatest#1 at ./essentials.jl:790 [inlined]
 [15] invokelatest at ./essentials.jl:789 [inlined]
 [16] eventloop(::ZMQ.Socket) at /Users/maria/.julia/packages/IJuli
a/F1GUo/src/eventloop.jl:8
 [17] (::getfield(IJulia, Symbol("##15#18")))() at ./task.jl:268
```

In [106]:

```
# Save the relevant arrays of the Lambert record in a .jld2 file
@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Lambert.jld2" uivD_L uivD
_L_EDC
```

vi) Binned resampling on grid

In [107]:

```
@load "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/Lambert.jld2"
```

Out[107]:

```
2-element Array{Symbol,1}:
 :uivD_L
 :uivD_L_EDC
```

- For general analyses, bin on 1 kyr timestep

In [131]:

```
binsize = 1 # each timestep is 1000 years
tmin_L = ceil(minimum(intpD_t_L))
tmax_L = floor(maximum(intpD_t_L))
grid_L = tmin_L + binsize/2 : binsize : tmax_L - binsize/2 # these must be the b
in edges if the bin midpoints shall be at every whole 1 kyr
```

Out[131]:

```
-799.5:1.0:-12.5
```

In [ ]:

```

```

In [120]:

```
resampling_method_L = BinnedResampling(grid_L, 1000) # resample 1000 draws (with
substitution) in each bin of the grid

@time L_binned_full = resample(uivD_L, resampling_method_L)
```

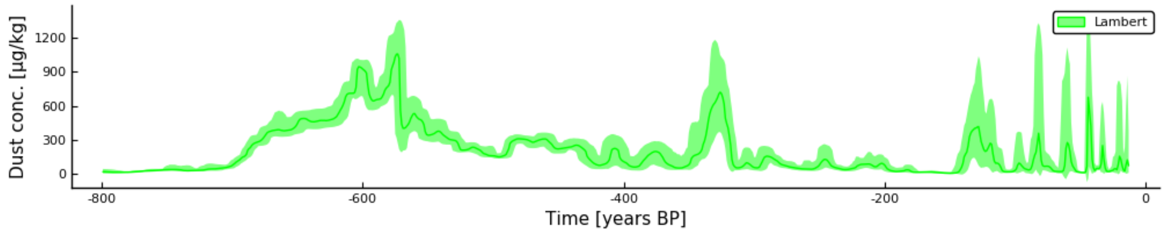
Out[120]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 787 uncertain values coupled with 787 uncertain indic
es
```

In [122]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
L = L_binned_full

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

# time array
binmidpoints_L = [L.indices[i].value for i in 1:length(L)]


plot_L_binned =
plot(size = (1000,200),
    binmidpoints_L, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :lime,
    label = "Lambert",
    xlabel = "Time [years BP]",
    ylabel = "Dust conc. [μg/kg]",
    grid = false
    )
```

Out[122]:



For analysis with the Bereiter record, which is from the same ice core of Epica Dome C, we prepare a version small age uncertainty (uncertainty from lock-in depth co gas bubbles in ice)

In [125]:

```
@time L_binned_full_EDC = resample(uivD_L_EDC, resampling_method_L) # version wi
th small age uncertainty
```

```
 39.533399 seconds (38.53 M allocations: 27.983 GiB, 16.77% gc time)
```

Out[125]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 787 uncertain values coupled with 787 uncertain indic
es
```

- For higher resolution analyses, we prepare a finer grid on which to bin the time series.

In [189]:

```
binsize_hr125 = 0.125 # 125 year bins, for hr analysis with time step like the r
esolution of the Grant hr record
grid_L_hr125 = tmin_L - binsize_hr125/2 : binsize_hr125 : tmax_L + binsize_hr125
/2 # bin edges, for the bin midpoints to start and end at whole 1 kyrs
```

Out[189]:

```
-800.0625:0.125:-11.9375
```

In [191]:

```
resampling_method_L_hr125 = BinnedResampling(grid_L_hr125, 1000)

@time L_binned_full_hr125 = resample(uivD_L, resampling_method_L_hr125)
```

```
224.662006 seconds (234.18 M allocations: 216.257 GiB, 20.17% gc tim
e)
```

Out[191]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 6305 uncertain values coupled with 6305 uncertain ind
ices
```

Since the mean resolution was a littlebit below the interplated hr resolution, we also prepare a hr version with 500 yr resolution, to see if these give any different results in analyses.

- prepare version for analyses with time step of 500 years

In [192]:

```
binsize_hr500 = 0.5 # time steps like the Bereiter hr record
grid_L_hr500 = tmin_L + 0.5/2 : 0.5 : tmax_L - 0.5/2
resampling_method_L_hr500 = BinnedResampling(grid_L_hr500, 1000)
```

Out[192]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1575 uncertain values coupled with 1575 uncertain ind
ices
```

In [196]:

```
@time L_binned_full_hr500 = resample(uivD_L, resampling_method_L_hr500)
```

Out[196]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1575 uncertain values coupled with 1575 uncertain ind
ices
```

- Prepare version with smaller age uncertainty for higher resoltuion analysis (time step 500 years) with the Bereiter pCO2 EDC record.

In [195]:

```
@time L_binned_full_hr500_edc = resample(uivD_L_EDC, resampling_method_L_hr500)
```

```
67.615267 seconds (62.83 M allocations: 54.653 GiB, 22.72% gc time)
```

Out[195]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1575 uncertain values coupled with 1575 uncertain ind
ices
```

Save all the binned resampled versions of the Lambert time series in a .jld2 file

In [197]:

```
# save all the binned resampled versions of the Lambert time series
@save "../Koding/WrangledDataFiles/Binned_ts_fullength/Lambert.jld2" L_binned_fu
ll L_binned_full_EDC L_binned_full_hr125 L_binned_full_hr500 L_binned_full_hr500
_edc
```

In [200]:

```
# check that they are all there
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/Lambert.jld2"
```

Out[200]:

```
5-element Array{Symbol,1}:
 :L_binned_full
 :L_binned_full_EDC
 :L_binned_full_hr125
 :L_binned_full_hr500
 :L_binned_full_hr500_edc
```

4.2 - Martinez-Garcia

- 4.2 Ma marine sediment core of dust (Fe-proxy)
- Data available from Pangaea. DOI: https://doi.org/10.1594/PANGAEA.767460 (https://doi.org/10.1594/PANGAEA.767460). Supplement of article from Martinez-García et al. (2011).

In [1]:

```
# Read in Martinez-Garcia dataset

filepath_MG = "../../MASTER_2.0/data/dust/Martinez-Garcia_2011/datasets/177-1090
_Fe_dust_acc.tab"
rawD_MG = readdlm(filepath_MG, skipstart = 1, dims = (6165,3))

# name columns
age_MG_    = rawD_MG[:,1]        # Age [ka BP]
Fe_MG_   = rawD_MG[:,2] / 1000  # Accumulation rate Fe [mg/m**2/a]      # * 1/10
00 gives SI-units [g/m^2/year]
dust_MG_ = rawD_MG[:,3] / 10    # Accumulation rate dust [g/cm**2/ka]   # * 1/10
gives SI-units [g/m^2/year]
;

plot(xlabel = "Age [ka BP]", ylabel = string("Accumulation rate ",L"[mg/m^2/a]"
))
plot(age_MG_, Fe_MG_,  label = "Fe")
plot!(age_MG_, Fe_MG_, label = "dust")
```

```
UndefVarError: readdlm not defined

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

ii) Redefine age as time

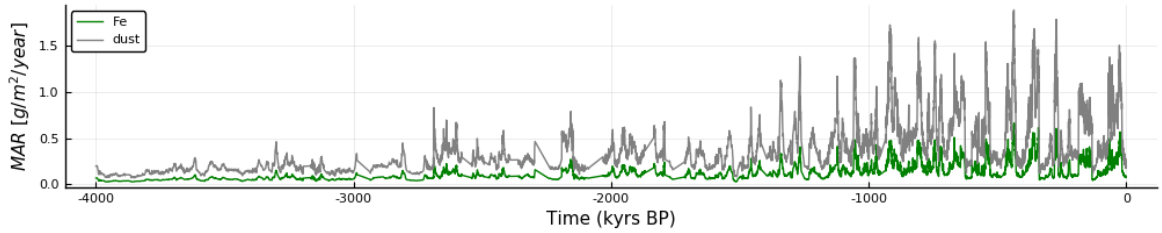
In [125]:

```
# reverse the arrays
t_MG    = -reverse(age_MG_)    # negative to increase towards present (defined a
s 0)
Fe_MG   = reverse(Fe_MG_)      # Accumulation rate Fe [g/m^2/year]
dust_MG = reverse(dust_MG_)    # Accumulation rate dust  [g/m^2/year]
;
```

In [227]:

```
# plot Martinez-Garcia dust and Fe record

plot_MG_FeDust =
plot(xlabel = "Time (kyrs BP)",
    ylabel = L"MAR \ [g/m^2/year]",
    size = (1000,200))
plot!(t_MG, Fe_MG,
    color = :green,
    label = "Fe")
#savefig("../figurar/RawData/Dust/Martinez-Garcia_Fe.pdf")
plot!(t_MG, dust_MG,
    color= :grey,
    label = "dust")
#savefig("../figurar/RawData/Dust/Martinez-Garcia_FeDust.pdf")
```

Out[227]:



Uncertainties

Uncertainties was not included in the dataset, but was reported in the article as following: "After error propagation, we find that the analytical component of the uncertainty is 7.8% [...] of the final value (1σ) for Fe [...]" (Martinez-García et al. (2011), under Methods summary).

In [126]:

```
# Analytical uncertainty
Fe_1σ_MG = Fe_MG .* (0.078)      # 1σ = 7.8% of final value for Fe
dust_1σ_MG = dust_MG .* (0.084) # 1σ = 8.4% of final value for Ti (we don't incl
ude this record) (dust)
;
```

Potential systematic deviations in the age model, as reported in article:

Maximum age model envelope:

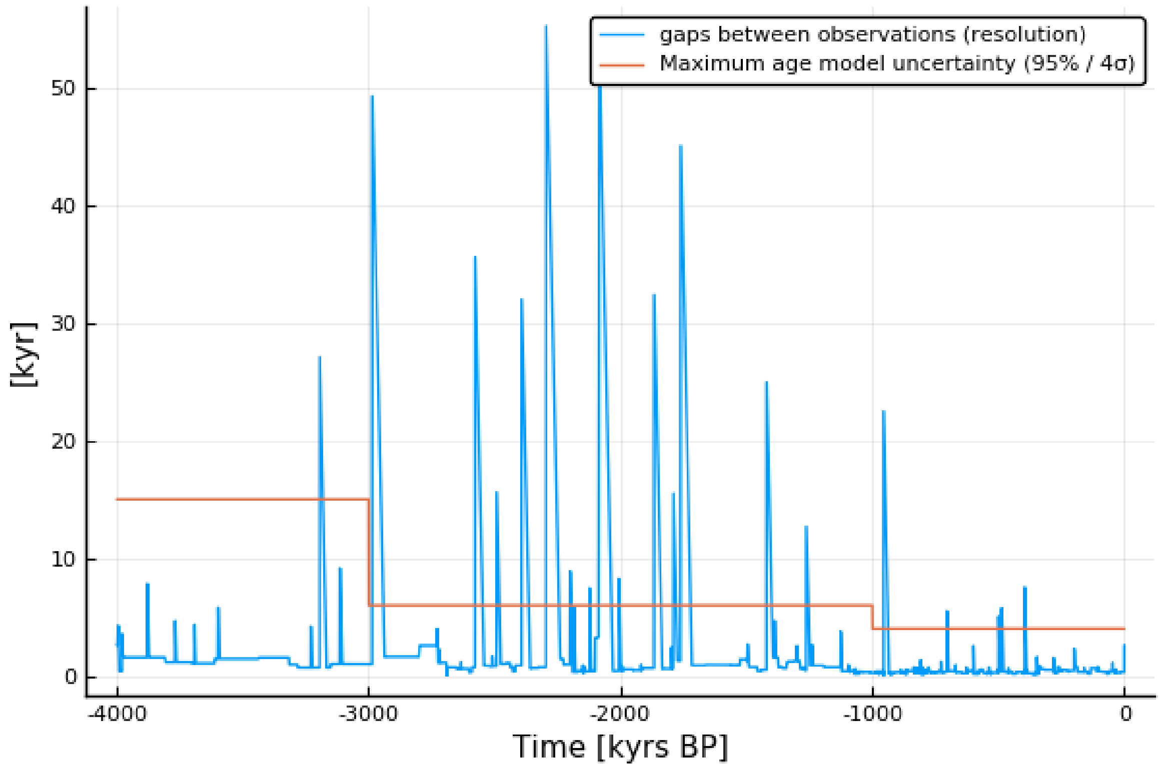
- 4 kyr for the interval 0-1 Myr ago
- 6 kyr for the interval 1-3 Myr ago
- 15 kyr for the interval 3-4 Myr ago

We *interpret* "maximum age model envelope" as the 95% confidence interval, that is 4σ ($\pm 2\sigma$). We should therefore divide by 4 to get 1σ.

In [129]:

```
# Potential systematic deviations in age model

# Maximum age model envelope of 4 kyr for the interval 0-1 Myr ago # (we interpr
et this as  +- 2σ = 4σ)
# Maximum age model envelope of 6 kyr for the interval 1-3 Myr ago
# Maximum age model envelope of 15 kyr for the interval 3-4 Myr ago
t_1σ_MG = zeros(length(t_MG));

t_1σ_MG[t_MG .> -1000] .= 4/4                      # 4σ = 4 kyrs, so divide by
 4 to get 1σ
t_1σ_MG[(t_MG .<= -1000) .& (t_MG .> -3000)] .= 6/4
t_1σ_MG[t_MG .<= -3000] .= 15/4

plot(t_MG, diff(t_MG), label = "gaps between observations (resolution)")
plot!(t_MG, 4*t_1σ_MG, # Showing stepwise definition of age uncertainty, good.
    xlabel = "Time [kyrs BP]",
    ylabel = "[kyr]",
    label = "Maximum age model uncertainty (95% / 4σ)")
```

Out[129]:



Save the arrays

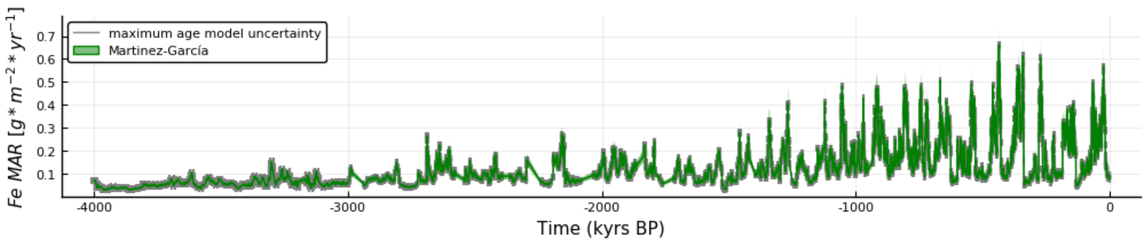
In [232]:

```
@save "../Koding/WrangledDataFiles/BasicArrays/MartinezGarcia.jld2" t_MG t_1σ_MG
Fe_MG Fe_1σ_MG
```

Plot Fe MAR record with value andcertainty and age model uncertainty

In [130]:

```
# plot of only Fe record
@load "../Koding/WrangledDataFiles/BasicArrays/MartinezGarcia.jld2"

plot_Fe_MG_ageunc =
plot(xlabel = "Time (kyrs BP)",
    ylabel = L"Fe \ MAR \ [g*m^{-2}*yr^{-1}]", # flux = accumulation rate (SYNON
YMS?)
    size = (1000,200))
plot!(t_MG, Fe_MG, xerr = 2 * t_1σ_MG, color = :grey, label = "maximum age model
uncertainty") # age uncertainty (95% confidence interval)
plot!(t_MG, Fe_MG,
    ribbon = (2 * Fe_1σ_MG, 2 * Fe_1σ_MG), #  analytical uncertainty (95% confid
ence interval)
    color = :green,
    label = "Martinez-García",
    #label = "Martinez-García et al. (2011)",
    )
savefig("../figurar/RawData/Dust/MG_Fe_noageunc.pdf")
```

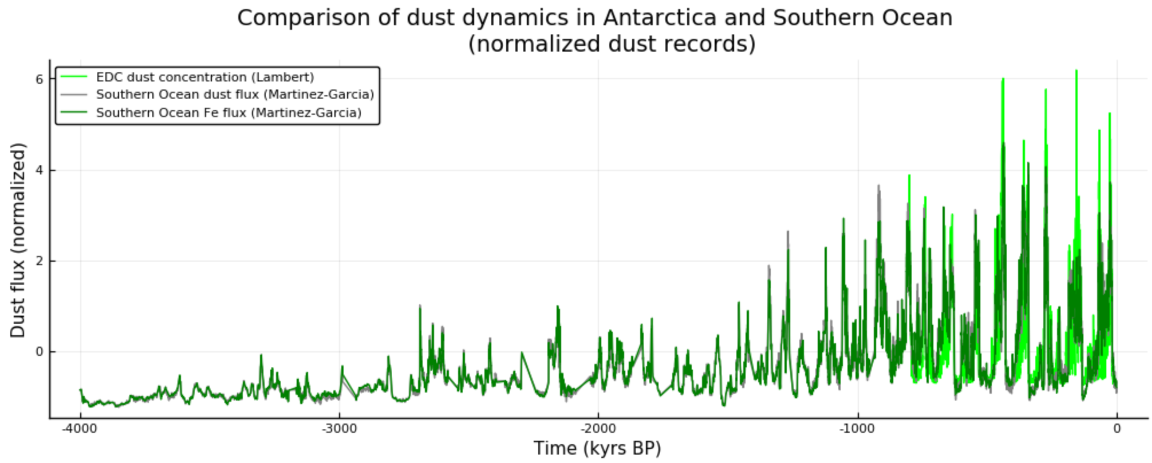


Normalize and plot Lambert and Martinez-Garcia **dust records to compare dynamics**:

In this normalized plot, we see that, even though the signal strength differs between the sites, the dynamics are the same for wind-born dust in the Southern Ocean (Martinez-Garcia marine core) and on East Antarctica (Lambert EDC ice core).

In [78]:

```
# Plot Lambert and Martinez-Garcia to compare

# normalize first, to avoid cluttering denominations and keep only dynamical inf
ormation
norm_Fe_MG   = (Fe_MG .- mean(Fe_MG)) / std(Fe_MG)
norm_dust_MG = (dust_MG .- mean(dust_MG)) / std(dust_MG)
norm_dust_L  = (dust_L .- mean(dust_L)) /std(dust_L)


# plot
plot(title = "Comparison of dust dynamics in Antarctica and Southern Ocean
    (normalized dust records)",
    xlabel = "Time (kyrs BP)",
    ylabel = "Dust flux (normalized)",
    size = (1000, 400)
    )
    # twinx(), # HOW TO GIVE A SECOND Y-AXIS LABEL TO THE RIGHT?
    # not needed if we normalize the data
plot!(t_L, norm_dust_L,
    color = :lime,
    label = "EDC dust concentration (Lambert)"
    )
plot!(t_MG, norm_dust_MG,
    color = :grey,
    label = "Southern Ocean dust flux (Martinez-Garcia)"
    )
plot!(t_MG, norm_Fe_MG,
    color = :green,
    label = "Southern Ocean Fe flux (Martinez-Garcia)"
    )

savefig("../../Master_2.0/figurar/plot_dust_comparative_normalized.pdf")
```



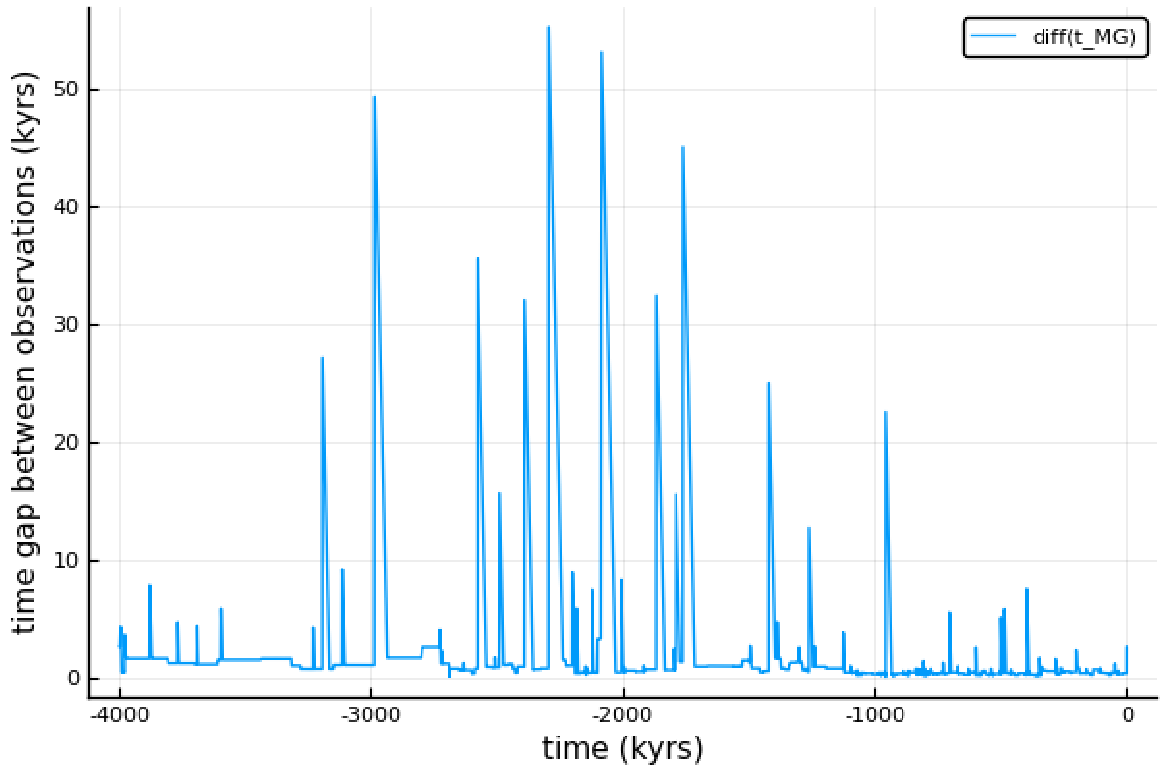
Notes on resolution

Is interpolation needed for the Martinez-Garcia data? We need at least one observation per 1000 years for our grid. Let's check the resolution of this time series:

In [120]:

```
# check resolution

minimum(diff(t_MG)) # 0.03 kyrs   # 30 years
mean(diff(t_MG))    # 0.650 kyrs  # 650 years
maximum(diff(t_MG)) # 55 kyrs

plot_MG_resolution =
plot(t_MG, diff(t_MG),
    xlabel = "time (kyrs)",
    ylabel = "time gap between observations (kyrs)",
    label = "diff(t_MG)",
)
```

Out[120]:



There are large variations in resolution of the marine dust record from Martinez-García. There are gaps of several tens of thousands of years at places, which call for interpolation in order to run analyses on millenial scale.

However, it is *less clear what resolution we should give our interpolated data*. On average the record has a high resolution, up to decadal resolution in parts, and with an average of one observation per 640 years.

> Will it be acceptable to interpolate to for example a centennial resolution and run high resolution analyses?
>
> We may later determine this by a *senistivity analysis*, testing if we get different results using different interpolation grids.

Interpolation

In [84]:

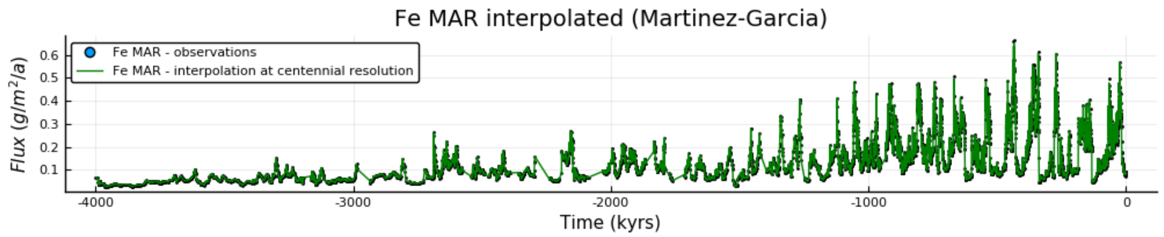
```
# interpolation

# define functions to interpolate
t_MG_interpolate = LinearInterpolation(t_MG, t_MG)
t_1σ_MG_interpolate = LinearInterpolation(t_1σ_MG, t_MG)
Fe_MG_interpolate = LinearInterpolation(Fe_MG, t_MG);
Fe_1σ_MG_interpolate = LinearInterpolation(Fe_1σ_MG, t_MG)

# defining a fine-grained grid that will be used to collect the interpolated val
ues
fine_grid_MG = ceil(minimum(t_MG)) : 0.1 : floor(maximum(t_MG)) # one value per
 100 years.
print(fine_grid_MG)

# we then put the interpolated values in a new object interpD_MG, one value in e
very bin of the fine time grid
intpD_t_MG = [t_MG_interpolate(i) for i in fine_grid_MG]
intpD_t_1σ_MG = [t_1σ_MG_interpolate(i) for i in fine_grid_MG]
intpD_Fe_MG = [Fe_MG_interpolate(i) for i in fine_grid_MG]
intpD_Fe_1σ_MG = [Fe_1σ_MG_interpolate(i) for i in fine_grid_MG]
;
```

```
-4000.0:0.1:-1.0
```

In [89]:

```
# check interpolation by plotting

plot(title = "Fe MAR interpolated (Martinez-Garcia)",
    xlabel = "Time (kyrs)",
    ylabel = L"Flux \ (g/m^2/a)",
    size = (1000,200))
scatter!(t_MG, Fe_MG,
    label = "Fe MAR - observations",
    markersize = 1
    )
plot!(intpD_t_MG, intpD_Fe_MG,
    #ribbon = (intpD_Fe_1σ_MG),
    label = "Fe MAR - interpolation at centennial resolution",
    color = :green
    )
```

Out[89]:



NB! it appears some observations are not captured by the interpolation at millennial scale....

> Question: From a previous plot (hidden above) it appeared that Some *observations were not captured by the interpolation*, not even when decreasing binsize to year scale (data is given on decade scale). I cannot see if that's the case from this coarse plot, but I only have low-resolution visuals. How can I control that this is not still the case ?

Redefining the interpolated dataset as an uivD

In [90]:

```
# redefining the interpolated Fe record intpD_MG as an uivD

t_uiv_MG = [UncertainValue(Normal, intpD_t_MG[i], intpD_t_1σ_MG[i]) for i in 1:length(intpD_t_MG)]  # Potential deviations in age model, as reported in article
Fe_uiv_MG = [UncertainValue(Normal, intpD_Fe_MG[i], (0.078.*intpD_Fe_MG[i])) for i in 1:length(intpD_Fe_MG)] # 1σ = 7.8%, reported in article.
uivD_FeMG = UncertainIndexValueDataset(t_uiv_MG, Fe_uiv_MG)
```

Out[90]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDataset} containing 39991 uncertain values coupled with 39991 uncertain indices
```

Save the relevant arrays in a .jld2 file

In [91]:

```
# Save the relevant arrays of the Martinez-García record in a .jld2 file

@save "../../MASTER_2.0/Koding/WrangledDataFiles/uivDs/MartinezGarcia.jld2" uivD_FeMG t_MG
```

vi) Binned resampling

In [6]:

```
@load "../Koding/WrangledDataFiles/uivDs/MartinezGarcia.jld2"
```

Out[6]:

```
3-element Array{Symbol,1}:
 :intpD_t_MG
 :uivD_FeMG
 :t_MG
```

In [94]:

```
# Binned resampling  for the main 1 kyr grid
binsize = 1
tmin_MG = ceil(minimum(t_MG))
tmax_MG = floor(maximum(t_MG))
grid_MG = tmin_MG + binsize/2: binsize : tmax_MG - binsize/2

resampling_method_MG = BinnedResampling(grid_MG, 1000)

@time MG_binned_fullength = resample(uivD_FeMG, resampling_method_MG)

@save "../../MASTER_2.0/Koding/WrangledDataFiles/Binned_ts_fullength/MartinezGar
cia.jld2" MG_binned_fullength
```

```
727.512050 seconds (685.45 M allocations: 691.831 GiB, 24.55% gc tim
e)
```

In [7]:

```
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/MartinezGarcia.jld2"
```

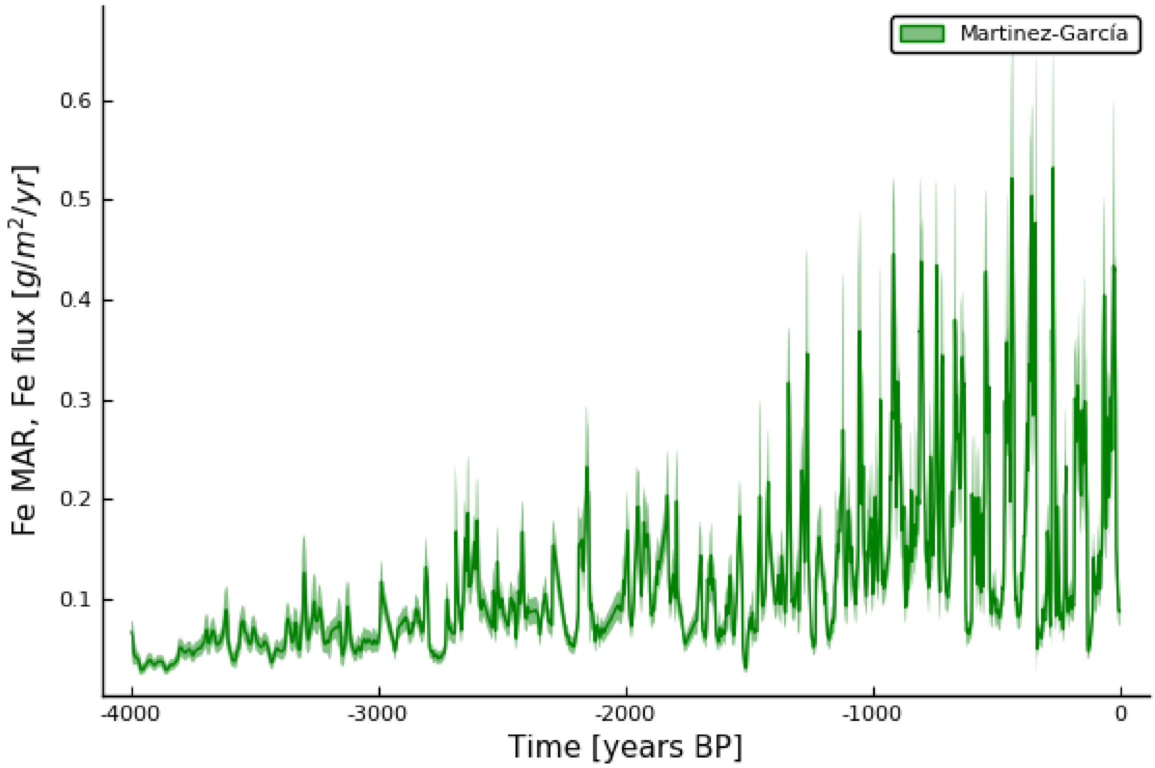
Out[7]:

```
1-element Array{Symbol,1}:
 :MG_binned_fullength
```

In [96]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
MG = MG_binned_fullength

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

# time array
binmidpoints_MG = [MG.indices[i].value for i in 1:length(MG)]


plot_MG_binned =
plot(binmidpoints_MG, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :green,
    label = "Martinez-García",
    xlabel = "Time [years BP]",
    ylabel = string("Fe MAR, Fe flux ", L"[${g/m^{2}}/yr$]"),
    grid = false
    )
```

Out[96]:



--

Points of discussion

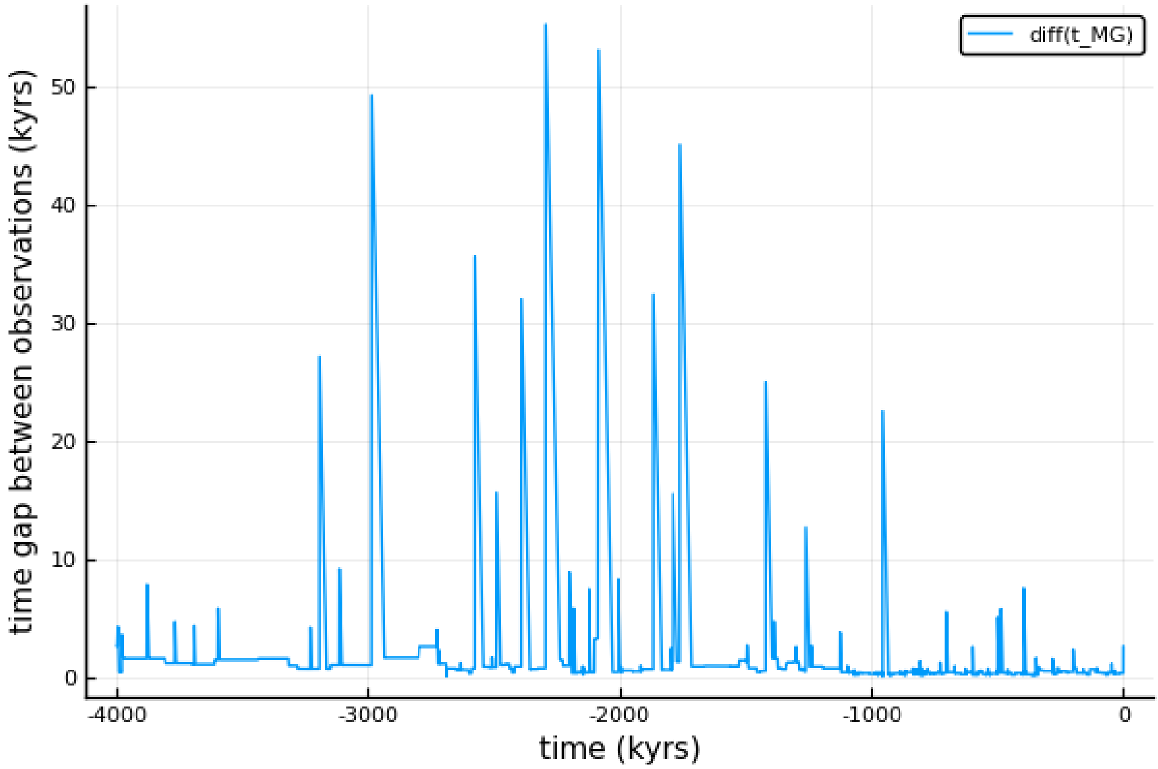
- SHOULD we do hr analysis of this record?
- if so, what is a reasonable BINSIZE for hr analysis?

In [121]:

```
print("Mean resolution of the Martinez-García record is ", mean(diff(t_MG)), "
 (650 years).
Highest resolution of the Martinez-García record is ", minimum(diff(t_MG)), " (3
0 years).
largest time gap is ", maximum(diff(t_MG)), " (55 000 years)." )

plot_MG_resolution
```

```
Mean resolution of the Martinez-García record is 0.6488822193380921
(650 years).
Highest resolution of the Martinez-García record is 0.029999999999997
2715 (30 years).
largest time gap is 55.2199999999998 (55 000 years).
```

Out[121]:



Is the record suitable for high resolution analysis?

Even though Martinez-Garcia record is a high resolution record at the baseline, the large time gaps represent a problem for using the record for high resolution analysis. However, we notice that the first part of the record (post-MPT) has smaller time gaps and overall higher resolution. Perhaps we can run higher resolution analyses for selected time intervals of the record? Let's check.

The other records suitable for high resolution analysis are the Grant GSL record, the La2004 record and the Chalk syn-MPT pCO2 record *(and possibly Lambert and Bereiter?)*. We therefore want to have a closer look at the resolution for the time intervals overlapping with these records.

1. From the plot above, the **post-MPT time interval** does seem to have high resolution for the Martinez-García record. Let's check the resolution for the relevant time interval.

Mostly, the resolution is just above a datapoint per 500 years.

In [8]:

```
#common time interval for the Grant and Martinez-García records

tmin_short = -800
tmin_G = -491 # recall for now, since haven't read in Grant data in this iterati
on
tmax_short = 0
tmin_MG = t_MG[1]
tmax_MG = t_MG[end]

maximum([tmin_short, tmin_MG]) : minimum([tmax_short, tmax_MG])
```

Out[8]:

```
-800.0:1.0:-1.0
```

In [31]:

```
# Cut the relevant interval from the MG record
t_MG_short = t_MG[(t_MG .> tmin_short)] # MG record for the past 800 kyrs

# make a plot to show the  resolution of the interval
plot(t_MG_short, diff(t_MG_short),
    label = "resolution",
    xlabel = "Time (kyrs BP)",
    xticks = (-800:100:0),
    ylabel = "time gaps [kyrs]",
    yticks = (0:1:8),
    title = "MG record resolution over the past 800 kyrs")
```

```
UndefVarError: t_MG not defined

Stacktrace:
 [1] top-level scope at In[31]:1
```

In [10]:

```
print("For the last 800 kyrs of the Martinez-García record,
mean resolution is ", mean(diff(t_MG_short)), " (330 years).
Highest resolution for the period is ", minimum(diff(t_MG_short)), " (140 year
s).
Largest time gap is ", maximum(diff(t_MG_short)), " (ca. 7500 years)." )
```

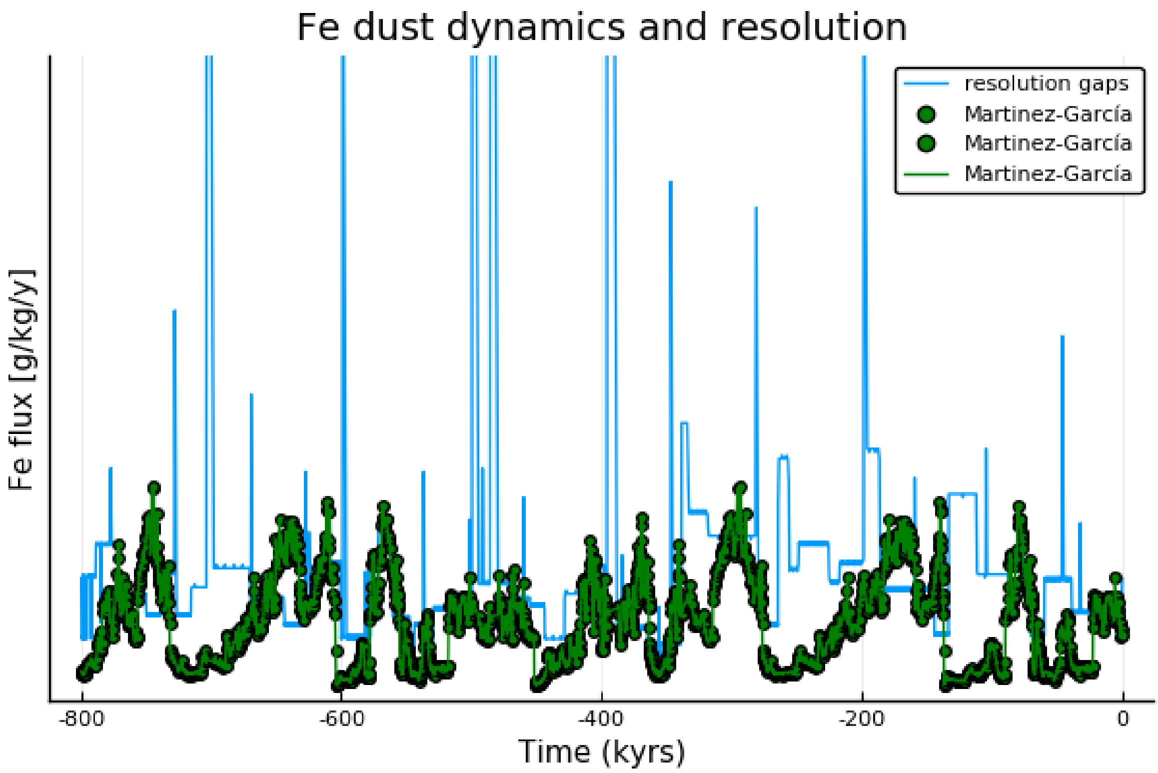
```
For the last 800 kyrs of the Martinez-García record,
mean resolution is 0.3290329218106996 (330 years).
Highest resolution for the period is 0.13999999999998636 (140 year
s).
Largest time gap is 7.519999999999982 (ca. 7500 years).
```

Is it acceptable to run high resolution analysis on this time interval? Let's first also check if the record in general has fast or slow dynamics.

In [19]:

```
# Cut out the time interval, and get the snippet of the Martinez-García Fe recor
d spanning the past 500 kyrs
Fe_MG_short = Fe_MG[(t_MG .> tmin_G)]


# plot resolution across the interval
#plot(t_MG_short, diff(t_MG_short), label = "resolution gaps", xlabel = "Time (k
yrs BP)", ylabel = "time gaps [kyrs]", title = "MG record resolution over past 5
00 kyrs")
# plot Fe dynamics across the interval
plot!(t_MG_short, Fe_MG_short,
    color = :green,
    xlabel = "Time (kyrs)",
    ylabel = "Fe flux [g/kg/y]",
    label = "Martinez-García",
    title = "Fe dust dynamics and resolution")
```

Out[19]:



I'd say the time series looks like it has *fast dynamics where it has high resolution, which does not speak in favour of interpolating for higher resolution analyses*.

> Input? Interpret this as fast dynamics, or "too small fluctuations to care about"?

But let's give it a try anyways with high resolution analysis for fun, but keep in mind the assumptions made when interpreting the results. We have more or less the same case with the Lambert time series, so that might add *some* robustness to the result.

binsize_hr = 0.125 (like Grant record) will be too fine. Let's stay above the mean resolution, and interpolate values for the high resolution grid to one value each 500 years.

> Can fake data from linear interpolation might bias the result to a causal connection ("precede and predict"))?

In [12]:

```
# common grid MG - G with hr binsize 0.5 kyrs
    # NB. this implies we have to make a version for Grant hr at 500 as well

binsize_hr500 = 0.5


grid_MG_hr500 = tmin_short - binsize_hr500/2 : binsize_hr500 : tmax_MG + binsize
_hr500/2 # to get values at binmidpoints, substract half a binsize from tmin, an
d add half a binsize to tmax
resamplingmethod_MG_hr500 = BinnedResampling(grid_MG_hr500, 1000)

@time MG_binned_postmpt_hr500 = resample(uivD_FeMG, resamplingmethod_MG_hr500)
# (191 seconds to run BinnedResampling on this grid)
```

```
282.095583 seconds (289.81 M allocations: 277.983 GiB, 19.30% gc tim
e)
```

Out[12]:

```
UncertainIndexValueDataset{UncertainIndexDataset,UncertainValueDatas
et} containing 1600 uncertain values coupled with 1600 uncertain ind
ices
```

In [84]:

```
# If we've binned correctly these should be the same:

print(tmin_short : tmax_MG, "
", MG_binned_postmpt_hr500.indices[1].value :  MG_binned_postmpt_hr500.indices[e
nd].value) # first and last binmidpoints of binned time series

#hurra!
```

```
-800.0:1.0:-1.0
-800.0:1.0:-1.0
```

In [20]:

```
@save "../Koding/WrangledDataFiles/Binned_ts_fullength/MartinezGarcia.jld2" MG_b
inned_fullength MG_binned_postmpt_hr500
```

In [22]:

```
# check that saved the 2 binned uivDs properly
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/MartinezGarcia.jld2"
# all good
```

Out[22]:

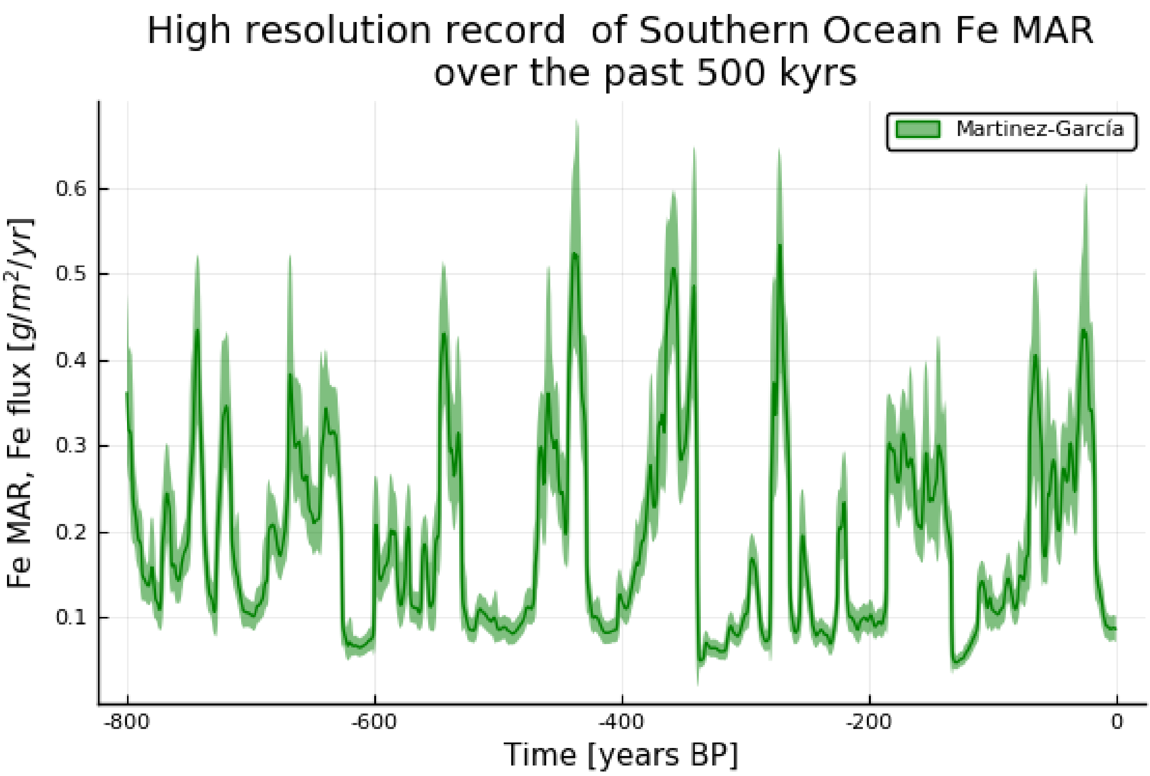
```
2-element Array{Symbol,1}:
 :MG_binned_fullength
 :MG_binned_postmpt_hr500
```

> *is 1000 draws enough to give a proper probability distribution?*

In [28]:

```
### Plot the binned resampled uivD time series with the 95% confidence interval
MG = MG_binned_postmpt_hr500

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

# time array
binmidpoints_MG = [MG.indices[i].value for i in 1:length(MG)]


plot_MG_binned =
plot(title = "High resolution record  of Southern Ocean Fe MAR
    over the past 500 kyrs",
    binmidpoints_MG, bin_median,
    ribbon = (bin_lower, bin_upper),
    color = :green,
    label = "Martinez-García",
    xlabel = "Time [years BP]",
    ylabel = string("Fe MAR, Fe flux ", L"[${g/m^{2}}/yr$]"),
    #xticks = (-800:0.5:0)
    )
```

Out[28]:



--

- **High resolution record syn-MPT**

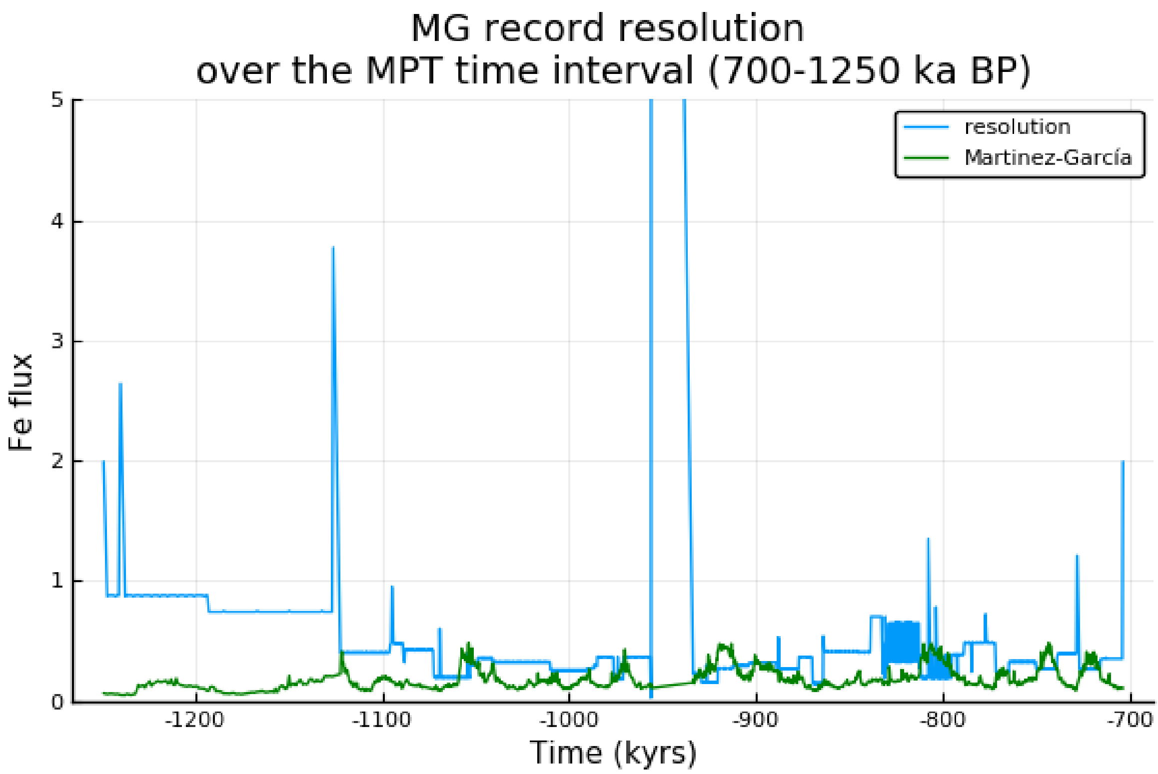
It would also be interesting to do a high resolution analysis of the dynamics between Fe dust, insolation and pCO2 **during the MPT**. The records available for this are Martinez-García, La2004 and Chalk. How is the resolution of the Martinez-García Fe MAR record across the MPT?

In [35]:

```
# MG record resolution during the MPT

# cut out the time interval of the MG record
t_MG_synMPT = t_MG[(t_MG .> -1250) .& (t_MG .< -700)] # relevant interval for an
alysis with La2004

print("For the syn-MPT time interval
Mean resolution is ", mean(diff(t_MG_synMPT)), " (350 years).
Smallest time gap is ", minimum(diff(t_MG_synMPT)), " (30 years).
Largest time gap is ", maximum(diff(t_MG_synMPT)), " (22.5 kyrs)." )


# plot to visualize resolution and if there are significant time gaps
plot(t_MG_synMPT, diff(t_MG_synMPT), ylims = (0,5), label = "resolution", xlabel
= "Time (kyrs BP)", ylabel = "time gaps [kyrs]", title = "MG record resolution
over the MPT time interval (700-1250 ka BP)")

#=Let's also check if the record in general has fast or slow dynamics
(if slow dynamics in the high resolution parts of the record, it might be defend
able to interpolate for a higher resolution). =#
Fe_MG_synMPT = Fe_MG[(t_MG .>-1250) .& (t_MG .<-700)]
plot!(t_MG_synMPT, Fe_MG_synMPT, color = :green, xlabel = "Time (kyrs)", ylabel
= "Fe flux", label = "Martinez-García") # title = "Fe dust dynamics 1.250 - 1.08
0 Ma BP")
```

```
For the syn-MPT time interval
Mean resolution is 0.37929117442668525 (350 years).
Smallest time gap is 0.029999999999972715 (30 years).
Largest time gap is 22.5 (22.5 kyrs).
```

Out[35]:



With a mean resolution of 380 years, we deem it reasonable to run high resolution analysis with the Martinez-García dust record over the syn-MPT, with a timestep to 0.5 kyrs.

- Let's check if the Martinez-García Fe MAR record has sufficiently high resolution for the time interval overlapping with the Chalk record (1080-1250 ka BP).

In [36]:

```
# cut out the time interval of the MG record overlapping with the Chalk pCO2 rec
ord
tmin_C = -1242
tmax_C = -1088

t_MG_synC = t_MG[(t_MG .>= tmin_C) .& (t_MG .<= tmax_C)]
t_MG_synC[1] : t_MG_synC[end]  # (does not concord because t_MG is not binned on
a regular time grid yet)
```

Out[36]:

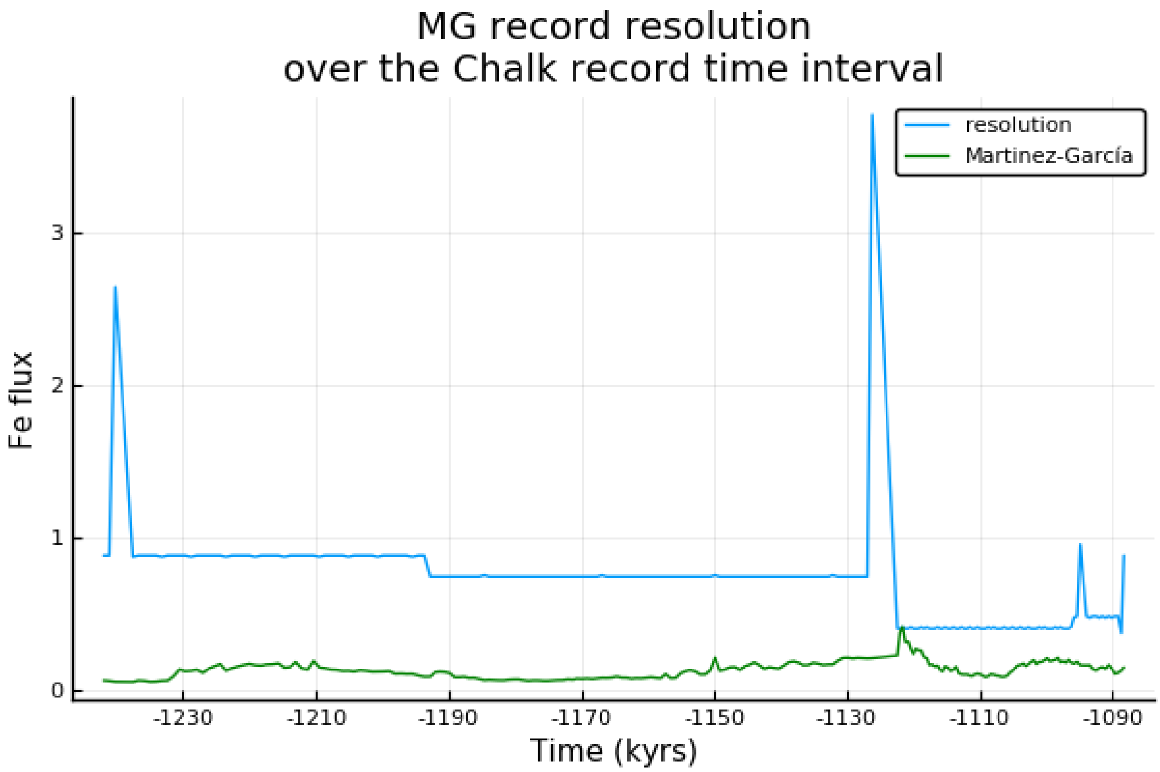
```
-1241.98:1.0:-1088.98
```

In [39]:

```
# MG record resolution for the Chalk record time interval

print("Mean resolution of the Martinez-García record over the Chalk record time
 interval is ", mean(diff(t_MG_synC)), " (680 years).
Highest resolution for the period is ", minimum(diff(t_MG_synC)), " (320 years).
Largest time gap is ", maximum(diff(t_MG_synC)), " (3.770 years)." )


# plot to visualize resolution
plot(t_MG_synC, diff(t_MG_synC), label = "resolution", xlabel = "Time (kyrs BP)"
, ylabel = "time gaps [kyrs]", title = "MG record resolution
over the Chalk record time interval")


#=Let's also check if the record in general has fast or slow dynamics
(if slow dynamics in the high resolution parts of the record, it might be defend
able to interpolate for a higher resolution). =#
Fe_MG_synC = Fe_MG[(t_MG .>= tmin_C) .& (t_MG .<= tmax_C)]
plot!(t_MG_synC, Fe_MG_synC, color = :green, xlabel = "Time (kyrs)", ylabel = "F
e flux", label = "Martinez-García", title = "Fe dust dynamics 1.250 - 1.080 Ma B
P")
plot!(title = "MG record resolution
over the Chalk record time interval")
plot!(xticks = (-1250:20:1080))
```

```
Mean resolution of the Martinez-García record over the Chalk record
time interval is 0.680132743362832 (680 years).
Highest resolution for the period is 0.37000000000011823 (320 year
s).
Largest time gap is 3.769999999999982 (3.770 years).
```

Out[39]:



We see that most of the record has a coarser resolution than 500 years for most of the MTP time interval (mean resolution of one value per 680 years). Only the very first part of the record has a 500 yr resolution (ca 1090-1120 kyrs BP). We could consider running a high resolution analysis over that time interval, but the time series length would unfortunately be too short to give robust results (need at least 100 datapoints, and this interval would give us only 30 kyrs/500 yrs = 60 datapoints). We therefore conclude that **the preMPT syn-Chalk time interval of the record is not suitable for a higher resolution analysis**. We will therefore run the analysis only on the 1 kyr timestep grid, for which the Martinez-Garcia record has sufficient resolution.

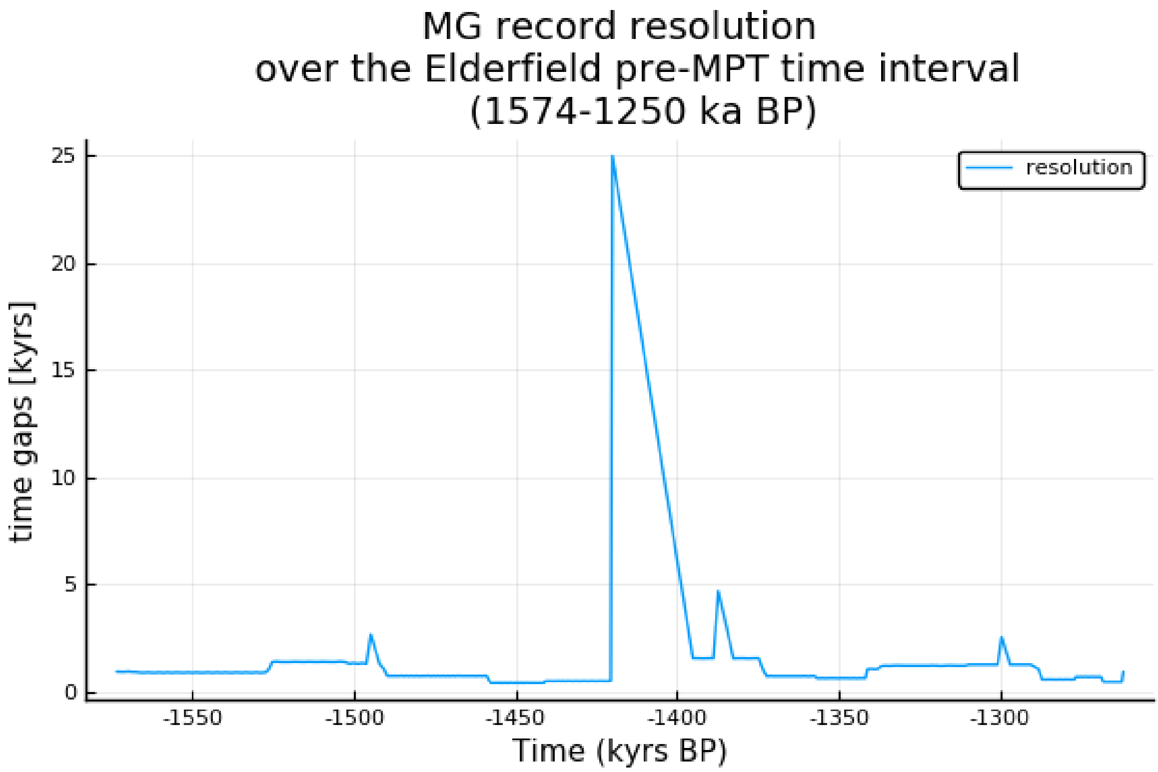
- Finally, let's check the resolution of the record for the pre-MPT which overlaps with the Elderfield record

In [53]:

```
# MG record resolution pre-MPT

# cut out the pre_MPT time interval of the MG record that overlaps with the Elde
rfield record
t_MG_preMPT_1500 = t_MG[(t_MG .> -1574) .& (t_MG .< -1250)] # relevant interval
 for analysis with La2004

print("Mean resolution of the Martinez-García record over the syn-MPT time inter
valis ", mean(diff(t_MG_preMPT_1500)), " (900 years).
Highest resolution for the period is ", minimum(diff(t_MG_preMPT_1500)), " (400
 years).
Largest time gap is ", maximum(diff(t_MG_preMPT_1500)), " (25 kyrs)." )


# plot to visualize resolution and if there are significant time gaps
plot(t_MG_preMPT_1500, diff(t_MG_preMPT_1500),
    label = "resolution",
    xlabel = "Time (kyrs BP)",
    ylabel = "time gaps [kyrs]",
    title = "MG record resolution
    over the Elderfield pre-MPT time interval
    (1574-1250 ka BP)")
```

```
Mean resolution of the Martinez-García record over the syn-MPT time
intervalis 0.8976945244956772 (900 years).
Highest resolution for the period is 0.40999999999998545 (400 years).
Largest time gap is 24.960000000000036 (25 kyrs).
```

Out[53]:



With a mean resolution of 900 years between each datapoint, and a time gap of 25 kyrs, this record is **not suitable for higher resolution analyses** over the pre-MPT time period. We will only use the standard 1 kyr grid already prepared.

- Finally, let's check the resolution of the record for the pre-MPT which overlaps with the Elderfield record

In [1]:

```
# MG record resolution pre-MPT

# cut out the pre_MPT time interval of the MG record that overlaps with the Elde
rfield record
t_MG_preMPT_4000 = t_MG[(t_MG .> tmin) .& (t_MG .< -1250)] # relevant interval f
or analysis with La2004

print("Mean resolution of the Martinez-García record over the syn-MPT time inter
valis ", mean(diff(t_MG_preMPT_4000)), " (900 years).
Highest resolution for the period is ", minimum(diff(t_MG_preMPT_4000)), " (400
 years).
Largest time gap is ", maximum(diff(t_MG_preMPT_4000)), " (25 kyrs)." )


# plot to visualize resolution and if there are significant time gaps
plot(t_MG_preMPT_4000, diff(t_MG_preMPT_4000),
    label = "resolution",
    xlabel = "Time (kyrs BP)",
    ylabel = "time gaps [kyrs]",
    title = "MG record resolution
    over the Elderfield pre-MPT time interval
    (1574-1250 ka BP)")
```

```
UndefVarError: t_MG not defined

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

Summary NB1

- We now have all the data on a regular time grid with time steps of 1 kyr. We have also some higher resolution regular grids, where we considered there was sufficient resolution in hte dataset for this.
- Associated uncertainties in time has been shuffled over to the value index, by resampling on the regular grid. The uncertainties carried on as kernel density estimates, which allows further resampling to estimate confidence intervals onwards.
- The wrangled data (the time series binned resampled on regular time grids) have been saved as .jld2 files, and can easily be imported in other notebooks for analysis.

The next step for the wrangled data will be to define a common time interval for the time series. Once the time series are on the same time grid (resolution and time interval), we can compute the predictive asymmetry from one to the other. This is done in the NBRs.

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

@save "../Koding/WrangledDataFiles/Binned_ts_fullength/ALL_ts_binned.jld2" LR04_binned_fullength_fullageunc LR04_binned_fullength_noageunc La2004_insol65N_fullength La2004_insol65N_fullength_hr SL_binned_fullength_ageunc SL_binned_fullength_noageunc E_binned_fullength_ageunc E_binned_fullength_noageunc G_binned_fullength G_binned_fullength_hr R_binned_full B_binned_fullength_ageunc B_binned_fullength_ageunc_hr B_binned_fullength_noageuncEDC B_binned_fullength_noageuncEDC_hr C_binned_1kyrgrid C_binned_hr L_binned_full L_binned_full_EDC L_binned_full_hr MG_binned_fullength MG_binned_postmpt_hr500 # missing L_binned_full_EDC_hr

In [54]:

```
@load "../Koding/WrangledDataFiles/Binned_ts_fullength/ALL_ts_binned.jld2"
```

Out[54]:

```
22-element Array{Symbol,1}:
 :LR04_binned_fullength_fullageunc
 :LR04_binned_fullength_noageunc
 :La2004_insol65N_fullength
 :La2004_insol65N_fullength_hr
 :SL_binned_fullength_ageunc
 :SL_binned_fullength_noageunc
 :E_binned_fullength_ageunc
 :E_binned_fullength_noageunc
 :G_binned_fullength
 :G_binned_fullength_hr
 :R_binned_full
 :B_binned_fullength_ageunc
 :B_binned_fullength_ageunc_hr
 :B_binned_fullength_noageuncEDC
 :B_binned_fullength_noageuncEDC_hr
 :C_binned_1kyrgrid
 :C_binned_hr
 :L_binned_full
 :L_binned_full_EDC
 :L_binned_full_hr
 :MG_binned_fullength
 :MG_binned_short_hr
```

save the wrangled data as a .jld2 file @save "../../MASTER_2.0/Koding/WrangledDataFiles/WrangledData.jld2" uivD_SL t_SL uivD_E intpD_t_E uivD_G t_G uivD_R t_R uivD_LR04 intpD_t_LR04 uivD_La2004 t_La2004 La2004 uivD_B uivD_B_EDC intpD_t_B uivD_C t_C uivD_L uivD_L_EDC intpD_t_L intpD_dust_L uivD_FeMG intpD_t_MG # This data can now be loaded in and read in any notebook, through the @load command: @load "/Users/maria/Dropbox/MASTER_2.0/Koding/WrangledData.jld2"

In [55]:

```
# This data can now be loaded in and read in any notebook, through the @load com
mand:
@load "/Users/maria/Dropbox/MASTER_2.0/Koding/WrangledData.jld2"
```

```
SystemError: opening file "/Users/maria/Dropbox/MASTER_2.0/Koding/Wr
angledData.jld2": No such file or directory

Stacktrace:
 [1] #systemerror#44(::Nothing, ::typeof(systemerror), ::String, ::B
ool) at ./error.jl:134
 [2] systemerror at ./error.jl:134 [inlined]
 [3] #open#311(::Bool, ::Bool, ::Bool, ::Bool, ::Bool, ::typeof(ope
n), ::String) at ./iostream.jl:289
 [4] #open at ./none:0 [inlined]
 [5] JLD2.MmapIO(::String, ::Bool, ::Bool, ::Bool) at /Users/maria/.
julia/packages/JLD2/hB4ya/src/mmapio.jl:100
 [6] openfile at /Users/maria/.julia/packages/JLD2/hB4ya/src/JLD2.j
l:194 [inlined]
 [7] #jldopen#9(::Bool, ::Bool, ::typeof(jldopen), ::String, ::Bool,
::Bool, ::Bool, ::Type{JLD2.MmapIO}) at /Users/maria/.julia/package
s/JLD2/hB4ya/src/JLD2.jl:231
 [8] jldopen at /Users/maria/.julia/packages/JLD2/hB4ya/src/JLD2.jl:
203 [inlined] (repeats 2 times)
 [9] #jldopen#10(::Base.Iterators.Pairs{Union{},Union{},Tuple{},Name
dTuple{(),Tuple{}}}, ::typeof(jldopen), ::String, ::String) at /User
s/maria/.julia/packages/JLD2/hB4ya/src/JLD2.jl:293
 [10] jldopen at /Users/maria/.julia/packages/JLD2/hB4ya/src/JLD2.j
l:288 [inlined] (repeats 2 times)
 [11] @load(::LineNumberNode, ::Module, ::Any, ::Vararg{Any,N} where
N) at /Users/maria/.julia/packages/JLD2/hB4ya/src/loadsave.jl:99
```

Recalling notations

The time arrays (prefix `t_` , or `intpD_t_` for interpolated datasets) contain the age values, formulated as negative values (kyrs from present). The data (prefix `uivD_`) contain the measured/reconstructed values, with their associated uncertainties carried on as kernel density estimates (KDE). The suffixes are abbreviations for the records, recalled here below:

- **LR04** = *Lisiecki & Raymo (2005)*. $\delta^{18}O$ global reference stack, spanning the last 5.3 Myr.
- **SL** = *Spratt & Lisiecki (2016)*. GSL stack (PCA of 5 different GSL records). Millennial resolution over the last 798 kyr.
- **E** = *Elderfield et al. (2012)*. GSL record, spanning the last 1.5 Myr (Mg/Ca temperature deconvolution of d18O).
- **G** = *Grant et al. (2014)*. Red Sea RSL record, spanning the last 500 kyr.
- **R** = *Rohling et al. (2014)*. Mediterranean RSL record, spanning the last 5.3 Myr
- **La2004** = Laskar (2004). Numerical solution for northern hemisphere summer insolation (top of atmosphere solar flux for 21st of June at 65°N). 50 Myr back and forth in time from present without significant uncertainty associated (uncertain beyond that).
- **B** = *Bereiter et al. (2015)*. pCO2 record from ice core at Epica Dome C, spanning ca 800 ka.
- **C** = *Chalk et al. (2017)*. pCO2 record from $\delta^{11}B$ proxy, spanning ca 1.090-1.240 Ma.
- **L** = *Lambert et al (2008)*. Dust record from dust concentration in ice core at Epica Dome C, spanning ca 800 kyr. (Note, the data has been set to updated age model since publication.)
- **MG** = *Martinez-Garcia et al. (2011)*. Record of iron mass accumulation rate (Fe MAR) in the Southern Ocean, spanning the last 4 Myr.

Overview plots of raw data

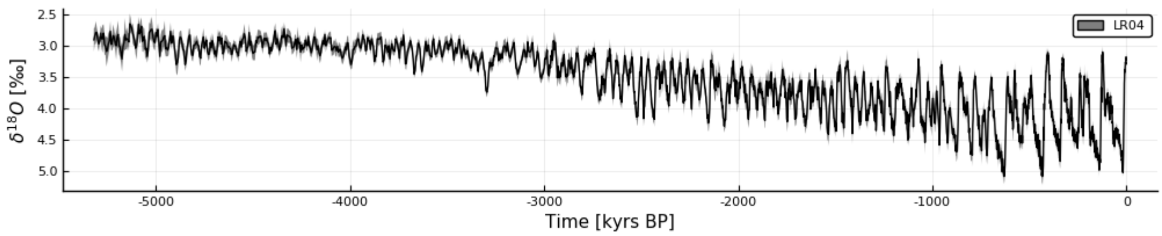
In [ ]:

```
@load "../data/sea-level/LR04/LR04_wrangled.jld2"
```

In [3]:

```
# LR04
@load "../data/sea-level/LR04/LR04_wrangled.jld2"
plot_LR04_noageunc = plot(
    t_LR04, d18O_LR04, ribbon = (2*d18O_1σ_LR04),
    xlabel = "Time [kyrs BP]", ylabel = L"$\delta^{18}O$ [$\perthousand$]",
    yflip = true, # by convention, we plot d18O on a reversed axis #(mirror read
ability of GSL, which constitutes on average 70% of the signal)
    color = :black, label = "LR04", size = (1000,200))
#plot!(t_LR04, d18O_LR04, yerr = 2*t_1σ_LR04) # 95% confidence interval of age m
odel uncertainty
```

Out[3]:



sea level records

gsl overview plot

In [ ]:

```
plot_gsl_overview =
    plot(plot_SL_binned, plot_E_binned, plot_G_binned, plot_R_binned,
    link = :x,
    layout = grid(4,1),
    xlim = (-2000,0)
)
```

In [ ]:

```
# Subplot of GSL records

#plot_GSL_overview =
plot(#title = "Global sea level records"
    plot_SprattLisiecki_raw_time,
    plot_Elderfield_raw_ageunc,
    plot_Grant_raw,
    plot_Rohling_raw,
    layout = (4,1),
    #share= :x,
    link = :x,
    size = (1000,400),
    xlims = (-1500:0), #(0:1500) # Why won't it cut at -1500 kyrs?
    ticks = false #(-1500:100:0)
    )
#savefig("../../Master_2.0/figurar/RawData/GSL/plot_GSL_overview.pdf")
```

gsl comparative plot

In [ ]:

```
# Plot to compare all GSL reconstructions

plot_GSL_comparative_full =
plot(#title = "Comparative plot of GSL records",
    xlabel = "Time [ka BP]",
    ylabel = "GSL [m]",
    #xlim = (-1000, 0),   # plot only a range 0:1000 ka
    ticks = (0:100:-1000),# not working
    grid= :true,          # not working
    size = (1000,200),
    share = :(x,y),       # working?
    #legend = :bottomleft
    )
plot!(t_R, RSL_mean_R,
    ribbon = (2*RSL_1σ_R), # ±2σ, aka 95% confidence interval
    color = :skyblue,
    label = "Rohling"
    )
#plot!(fine_grid_E, intpD_GSL_E, yerr = (2*t_1σ_E), color = :royalblue, ms = 0.
1) # LR04 age model uncertainty
plot!(fine_grid_E, intpD_GSL_E,
    ribbon = (2*intpD_1σ_E),
    color = :royalblue,
    label = "Elderfield"
    )
#plot!(t_SL, GSL_mean_SL, yerr = (2*t_1σ_SL), color = :darkblue, ms = 0.1) # LR0
4 age model uncertainty
plot!(t_SL, GSL_mean_SL,
    ribbon = (GSL_mean_SL - GSL_SL_err_lo, GSL_SL_err_up - GSL_mean_SL), # 95% C
I
    color = :darkblue,
    label = "Spratt & Lisiecki"
    )
plot!(t_G, RSL_G,
    ribbon = (RSL_G - RSL_q95lo_G, RSL_q95up_G - RSL_G),                 # 95% C
I
    color = :cyan,
    label = "Grant"
    )

savefig("../figurar/RawData/GSL/plot_comparative_GSL.pdf")
```

gsl - LR04 overview

In [ ]:

```
# Compare GSL records with d18O record

plot_overview_LR04_GSL =
plot(plot_LR04_raw_time,
    plot_GSL_comparative_full,
    layout = grid(2,1),
    link = :x)
savefig("../figurar/RawData/LR04/overview_LR04_GSL.pdf")
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```
# plot Lambert time series
# @load "../data/dust/Lambert_wrangled_raw.jld2"

plot_Lambert_time_noageunc =
plot(#title = "EDC aeolian dust record (Lambert lpc record)",
    size = (1000, 200),
    xlabel = "time (kyrs BP)",
    ylabel = "dust conc. [μg/kg]")
#scatter!(t_L, dust_L, xerr = 2 * t_1σ_L,  ms = 1, color = :grey, label  = "AICC
2012 age model uncertainty (2σ)")
plot!(t_L, dust_L,
    color = :lime,
    #label = "EDC dust record (Lambert)"
    label = "Lambert")

#savefig("../figurar/RawData/Dust/Lambert_aicc2012_noageunc.pdf")
```

In [ ]:

```

```

In [ ]:

```
# plot of only Fe record
@load "../data/dust/MG_raw.jld2"

plot_Fe_MG_noageunc =
plot(xlabel = "Time (kyrs BP)",
    ylabel = L"Fe \ MAR \ [g*m^{-2}*yr^{-1}]", # flux = accumulation rate (SYNON
YMS?)
    size = (1000,200))
#plot!(t_MG, Fe_MG, xerr = 2 * t_1σ_MG, color = :grey, label = "maximum age mode
l uncertainty") # age uncertainty (95% confidence interval)
plot!(t_MG, Fe_MG,
    ribbon = (2 * Fe_1σ_MG), #  analytical uncertainty (95% confidence interval)
    color = :green,
    label = "Martinez-García",
    #label = "Martinez-García et al. (2011)",
    )

#savefig("../figurar/RawData/Dust/MG_noageunc.pdf")
```

Normalize and plot Lambert and Martinez-Garcia dust records to compare

In [ ]:

```
# Plot Lambert and Martinez-Garcia to compare

# normalize first, to avoid cluttering denominations and keep only dynamical inf
ormation
norm_Fe_MG    = (Fe_MG .- mean(Fe_MG)) / std(Fe_MG)
norm_dust_MG = (dust_MG .- mean(dust_MG)) / std(dust_MG)
norm_dust_L  = (dust_L .- mean(dust_L)) /std(dust_L)

# We choose not to plot age uncertainty, not to clutter the plot


plot_dust_comparative_L_FeMG = #_wMGdust =
plot(#title = "Comparative plot of dust records",
    xlabel = "Time (kyrs BP)", ylabel = "Dust flux (normalized)", size = (1000,
200))
    # twinx(), # HOW TO GIVE A SECOND Y-AXIS LABEL TO THE RIGHT?# not needed if
 we normalize the data

# MG dust record (will not be used in analyses)
#plot!(t_MG, norm_dust_MG, color = :grey, label = "Martinez-García dust", # labe
l = "Southern Ocean dust flux (Martinez-Garcia)"
#    ribbon = (2*0.084*norm_dust_MG) )                                      # 1σ =
8.4%. We plot the 95% condifence interval (±2σ)

# MG Fe record
plot!(t_MG, norm_Fe_MG,  color = :green, label = "Martinez-García", # label = "S
outhern Ocean Fe flux (Martinez-Garcia)"
    ribbon = (2*0.078*norm_Fe_MG) )                                     # 1σ = 7.
8%. We plot the 95% condifence interval (±2σ)

# Lambert dust record
plot!(t_L, norm_dust_L, color = :lime, label = "Lambert" ) #label = "EDC dust co
ncentration (Lambert)")




#savefig("../figurar/plot_dust_comparative.pdf")
```

In this normalized plot, we see that, even though the signal strength differs between the sites, the dynamics are the same for wind-born dust in the Southern Ocean (Martinez-Garcia marine core) and on East Antarctica (Lambert EDC ice core).

In [ ]:

```
plot(plot_LR04_raw_time, # rerun above. Please, save the necessary stuff
    plot_GSL_comparative,
    plot_insolation, # rerun above. Please, save the necessary stuff
    plot_pCO2_comparative_B_C,
    #plot_pCO2_comparative_wHonisch,
    plot_dust_comparative_L_FeMG,
    #plot_dust_comparative_wMGdust
    layout = grid(5,1)
    )
```

Overview plots of wrangled time series

(interpolated and resampled on a regular grid)

In [2]:

```
#### d18O / sea level / ice volume proxies
@load "../WrangledDataFiles/Binned_ts_fullength/LR04.jld2" # LR04 - d18O stack
@load "../WrangledDataFiles/Binned_ts_fullength/SprattLisiecki.jld2" # Spratt&Lisiecki - global sea level stack, spanning last 800 kyrs (post-MPT)
@load "../WrangledDataFiles/Binned_ts_fullength/Elderfield.jld2" # Elderfield - global sea level record, spanning last 1.5 Myrs (pre- syn- & post-MPT)
@load "../WrangledDataFiles/Binned_ts_fullength/Grant.jld2" # Grant - Red Sea RSL record, spanning last 500 kyrs (post-MPT)
@load "../WrangledDataFiles/Binned_ts_fullength/Rohling.jld2" # Rohling - Mediterranean RSL record, spanning last 5.3 Myrs (pre- syn- & post-MPT)

#### insolation time series
@load "../WrangledDataFiles/La2004.jld2" # La2004 - numerical solution for northern hemisphere summer insolation, last 5 Ma computed using AnalySeries (Paillard, 1994). (pre- syn- & post-MPT).

#### pCO2 records/proxies
@load "../WrangledDataFiles/Binned_ts_fullength/Bereiter_nointp.jld2" # Bereiter - post-MPT pCO2 record - with age model uncertainty
@load "../WrangledDataFiles/Binned_ts_fullength/Bereiter_noageuncEDC_intp.jld2" # Bereiter - post-MPT pCO2 record - without age model uncertainty (for analysis with Lambert EDC dust record). NB. this time series containts interpolated values.
@load "../WrangledDataFiles/Binned_ts_fullength/Chalk.jld2" # Chalk - syn-MPT pCO2 record (time interval 1.088-1.242 Ma BP, updated cut to 1.092-1.240 Ma BP)

#### dust records
@load "../WrangledDataFiles/Binned_ts_fullength/Lambert.jld2" # Lambert - post-MPT Antarctic dust record (spanning last 800 kyrs)
@load "../WrangledDataFiles/Binned_ts_fullength/MartinezGarcia.jld2" # Martinez-García - 4 Ma Southern Ocean Fe dust record (pre- syn- & post-MPT)
;
```

```
Bereiter BinnedResampled without interpolation
```

Define the first and last time point in each record. These will be used in the NBRs to determine the common grid.

In [3]:

```
# Define the first and last time point in each record

##### d18O record

# LR04
LR04 = LR04_binned_fullength_fullageunc
# assign to tmin_LR04 the first time value in LR04 (that is, the LR04 record sta
rts at ``tmin_LR04`` years BP)
tmin_LR04 = LR04.indices[1].value  # the binned LR04 record starts at -802 kyrs
 BP (binmidpoint)
tmax_LR04 = LR04.indices[end].value # -0.3 kyrs BP ...last age value in the LR04
record (binmidpoint)
print("The LR04 d18O record spans from ", tmin_LR04, " to ", tmax_LR04, " kyrs B
P.")

##### insolation time series

# La2004 (Ins)
La2004 = La2004_insol65N_fullength
#= recall, insolation is not an uncertain index value dataset type.
We therefore use two arrays (time index and insolation value) =#
tmin_La2004 = La2004_t_fullength[1] # 5000
tmax_La2004 = La2004_t_fullength[end] # 0 kyrs BP (present day)
print("
The La2004 insolation time series (Ins) spans from ", tmin_La2004, " to ", tmax_
La2004, " kyrs BP.")

##### GSL records

# SprattLisiecki (SpraSL)
SL = SL_binned_fullength_ageunc
tmin_SL = SL.indices[1].value # -797.0 kyrs BP
tmax_SL = SL.indices[end].value # -1.0 kyrs BP
print("
Spratt Lisiecki  GSL stack (SpraSL) spans from ", tmin_SL, " to ", tmax_SL, " ky
rs BP.")

# Elderfield (EldSL)
E = E_binned_fullength_ageunc
tmin_E = E.indices[1].value # -1574 kyrs BP
tmax_E = E.indices[end].value # -8 kyrs BP
print("
The Elderfield GSL record (EldSL) spans from ", tmin_E, " to ", tmax_E, " kyrs B
P.")

# Grant (GraSL)
G = G_binned_fullength
tmin_G = G.indices[1].value # -491 kyrs BP
tmax_G = G.indices[end].value # -1 kyrs BP
print("
The Grant sea level record (GraSL) spans from ", tmin_G, " to ", tmax_G, " kyrs
 BP.")

# Rohling (RohSL)
R = R_binned_full
tmin_R = R.indices[1].value # -5329 kyrs BP
tmax_R = R.indices[end].value # -1.0 kyrs BP
print("
The Rohling sea level record (RohSL) spans from ", tmin_R, " to ", tmax_R, " kyr
```

```
s BP.")



##### pCO2 records

# Bereiter (BerCO2)
B = B_binned_fullength_ageunc
tmin_B = B.indices[1].value # -802 kyrs BP
tmax_B = B.indices[end].value # -3 kyrs BP
print("
The Bereiter pCO2 record (BerCO2) spans from ", tmin_B, " to ", tmax_B, " kyrs B
P.")

# Chalk (ChaCO2)
C = C_binned
tmin_C = C.indices[1].value # - 1242 kyrs BP
tmax_C = C.indices[end].value # -1088 kyrs BP
print("
The Chalk pCO2 record (ChaCO2) spans from ", tmin_C, " to ", tmax_C, " kyrs BP."
)

##### dust records

# Lambert (IceDust)
L = L_binned_full
tmin_L = L.indices[1].value # -799 kyrs BP
tmax_L = L.indices[end].value # - 2 kyrs BP
print("
The Lambert dust record (IceDust) spans from ", tmin_L, " to ", tmax_L, " kyrs B
P.")

# Martinez-García (MarFe)
MG = MG_binned_fullength
tmin_MG = MG.indices[1].value # -3999 kyrs BP (4 Myrs)
tmax_MG = MG.indices[end].value # -2 kyrs BP
print("
The Martinez-Garcia Fe flux record (MarFe) spans from ", tmin_MG, " to ", tmax_M
G, " kyrs BP.")
MG_hr = MG_binned_postmpt_hr500
tmin_MG_hr = MG_hr.indices[1].value # -3999 kyrs BP (4 Myrs)
tmax_MG_hr = MG_hr.indices[end].value # -2 kyrs BP
print("
The higher resolution part of the MG record spans from ", tmin_MG_hr, " to ", tm
ax_MG_hr, " kyrs BP.")
```

```
The LR04 d18O record spans 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.
```

Plot the time series

> (Move to NB1?)

Define the plots of binned resampled time series that we will use in analyse

In [46]:

```
######## Plot the La2004 insolation time series

plot_La2004 =
plot(La2004_t_fullength, La2004_insol65N_fullength,
    color = :orange,
    label = "Ins    ",
    xlabel = "Time [kyrs BP]",
    ylabel = L"Solar \ flux \ [W/m^{2}]",
    legend = :right,
    grid = false,
    size = (1000,200)
    )

######### plot the BinnedResampled LR04 time series
ts = LR04_binned_fullength_fullageunc

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

# computing the time series median in each bin (0.5 quantile), and the confidenc
e interval we want to use (95%)
bin_median = quantile.(ts.values, 0.5)
bin_upperq = quantile.(ts.values, 0.975) .- bin_median
bin_lowerq = bin_median .- quantile.(ts.values, 0.025)
;

plot_LR04 =
plot(binmidpoints_ts,
    bin_median,
    ribbon = (bin_lowerq, bin_upperq),
    fillalpha = 0.3,
    color = :black,
    label =  "LR04  ",
    xlabel =  "Time [kyrs BP]",
    ylabel =  L"\delta{18}O \ [\perthousand]",
    grid = false,
    yflip = true,
    size =(1000,200),
    legend = :right
)



############ plot the Spratt & Lisiecki binned resampled GSL time series
ts = SL_binned_fullength_noageunc
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_SL =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :darkblue,
    label =  "SpraSL",
```

```
    xlabel =  "Time [kyrs BP]",
    ylabel =  "GSL [m]",
    grid = false,
    size = (1000,200),
    legend = :bottomleft
    )

########## plot the Elderfield binned resampled GSL time series
ts = E_binned_fullength_ageunc
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]

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

plot_E =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :royalblue,
    label =  "EldSL ",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "GSL [m]",
    grid = false,
    size = (1000,200),
    legend = :bottomleft
    )

########## plot the Rohling binned resampled GSL time series
ts = R_binned_full
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]

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

plot_R =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :skyblue,
    label =  "RohSL ",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "GSL [m]",
    grid = false,
    size = (1000,200),
    legend = :right
    )

########## plot the Grant binned resampled GSL time series
ts = G_binned_fullength
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]

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

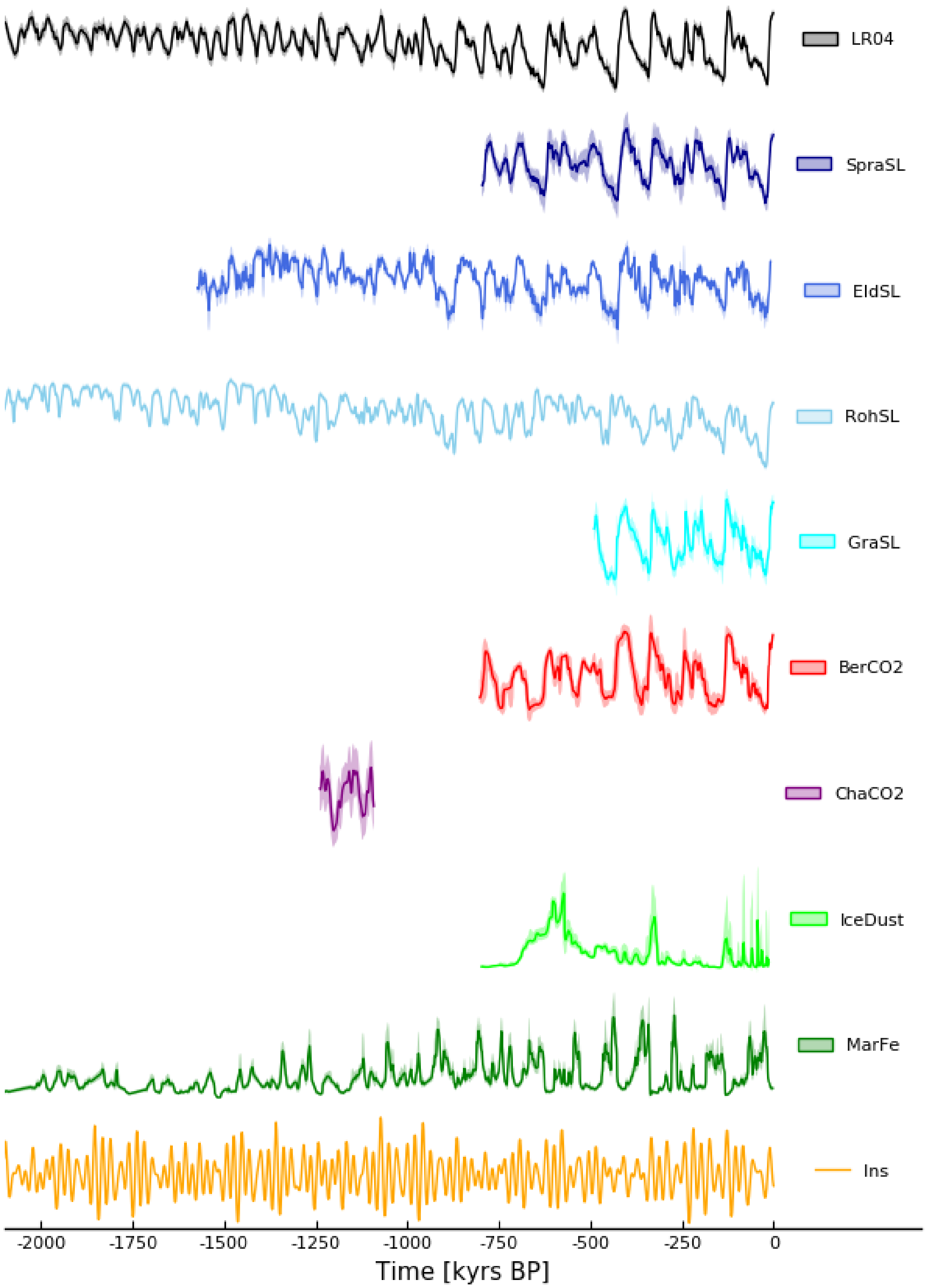
```
bin_lower = bin_median .- quantile.(ts.values, 0.025)

plot_G =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :cyan,
    label =  "GraSL ",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "GSL [m]",
    grid = false,
    size = (1000,200),
    legend = :bottomleft
    )

# plot the binned resampled Bereiter CO2 time series
ts = B_binned_fullength_ageunc
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_B =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :red,
    label =  "BerCO2",
    xlabel =  "Time [kyrs BP]",
    ylabel =  L"pCO{2} \ [ppmv]",
    grid = false,
    size = (1000,200),
    legend = :bottomleft
    )


# plot the binned resampled Bereiter CO2 time series - version without age uncer
tainty
ts = B_binned_fullength_noageuncEDC
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_B_edc =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :pink,
    label =  "BerCO2 - without age model uncertainty",
    xlabel =  "Time [kyrs BP]",
    ylabel =  L"pCO{2} \ [ppmv]",
    grid = false,
```

```
    size = (1000,200),
    legend = :left
    )


######### plot the binned resampled Chalk CO2 time series
ts = C_binned
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_C =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :purple,
    label =  "ChaCO2",
    xlabel =  "Time [kyrs BP]",
    ylabel =  L"pCO_{2} \ [ppmv]",
    grid = false,
    size = (1000,200),
    legend = :bottomleft
    )

######## Plot the two pCO2 time series in one figure

plot_pCO2 =
plot(xlabel = "Time [kyrs BP]", ylabel = L"pCO{2} \ [ppmv]", grid = false, size
= (1000,200), legend = :bottomleft)
#  Bereiter CO2
ts = B_binned_fullength_ageunc
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]
# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts.values, 0.5)
bin_upper = quantile.(ts.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts.values, 0.025)
plot!(binmidpoints_ts, bin_median, ribbon = (bin_lower, bin_upper), fillalpha =
0.3, color = :red, label =  "BerCO2")
# Chalk pCO2
ts = C_binned
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]
# computing the median in each bin (0.5 quantile), and the confidence interval w
e want to use (95%)
bin_median = quantile.(ts.values, 0.5)
bin_upper = quantile.(ts.values, 0.975) .- bin_median
bin_lower = bin_median .- quantile.(ts.values, 0.025)
plot!(binmidpoints_ts, bin_median, ribbon = (bin_lower, bin_upper), fillalpha =
0.3, color = :purple, label =  "ChaCO2")


######### plot the binned resampled Lambert dust time series
ts = L_binned_full
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]
```

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

plot_L =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :lime,
    label =  "IceDust",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc. [µg/kg]",
    grid = false,
    size = (1000,200),
    legend = :topleft
    )

######### plot the binned resampled Lambert dust time series with reduced age un
certainty
ts = L_binned_full_EDC
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_L_edc =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :olive,
    label =  "IceDust - age uncertainty between gas and ice",
    xlabel =  "Time [kyrs BP]",
    ylabel =  "Dust conc. [µg/kg]",
    grid = false,
    size = (1000,200),
    legend = :left
    )

######## plot the binned resampled Martinez-Garcia dust time series
ts = MG_binned_fullength
binmidpoints_ts =[ts.indices[i].value for i in 1:length(ts.indices)]


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

plot_MG =
plot(binmidpoints_ts, bin_median,
    ribbon = (bin_lower, bin_upper),
    fillalpha = 0.3,
    color = :green,
    label =  "MarFe ",
```

```
    xlabel =  "Time [kyrs BP]",
    ylabel =  string("Fe flux ", L"[g/m^{2}/yr]"),
    grid = false,
    size = (1000,200),
    legend = :topleft)
;

##################################################### OVERVIEW ALL TIME SERIES

po1 = plot(plot_LR04, xlabel = "", xaxis = :off)
po2 = plot(plot_SL, xlabel = "", xaxis = :off)
po3 = plot(plot_E, xlabel = "", xaxis = :off)
po4 = plot(plot_R, xlabel = "", xaxis = :off)
po5 = plot(plot_G, xlabel = "", xaxis = :off)
#po6 = plot(plot_pCO2, xlabel = "", xaxis = :off)
po6 = plot(plot_B, xlabel = "", xaxis = :off)
po7 = plot(plot_C, xlabel = "", xaxis = :off)
po8 = plot(plot_L, xlabel = "", xaxis = :off)
po9 = plot(plot_MG, xlabel = "", xaxis = :off)
po10 = plot(plot_La2004, xlabel = "Time [kyrs BP]", xaxis = :on)

#p_highlight_MPT = plot(Shape([-1250,-1250,-700,-700],[-100,100,100,-100]), fill
alpha = 0.1, color = :red, label = "MPT", ylims = (-10,10),
#        xlabel = "Time [kyrs BP]", xaxis = :on)

po_alltimeseries = plot(po1, po2, po3, po4, po5, po6, po7, po8, po9, po10, #p_hi
ghlight_MPT,
    size = (72*8.27, 72*11.69), # A4 (72 dots/inch)
    layout = grid(10,1),
    xlims = (-2100,400), xticks = (-5000:250:0),
    ylabel = "", yaxis = :off,
    legend = :right, bg_legend = :transparent,
    grid = :off,
    #Shape([-1250,-1250,-700,-700],[-100,100,100,-100]), fillalpha = 0.1, color
 = :red, label = "MPT",        # DOESN'T WORK
    )

#savefig("../../figurar/BinnedTimeseries/p_all_ts_A4.pdf")
```

Out[46]:


LR04
SpraSL
EldSL
RohSL
GraSL
BerCO2
ChaCO2
IceDust
MarFe
Ins
-2000
-1750
-1500
-1250
-1000
-750
-500
-250
0
Time [kyrs BP]

In [1]:

```
# Experiments with layout - highlight

p_highlight_MPT = plot(Shape([-1250,-1250,-700,-700],[-100,100,100,-100]), filla
lpha = 0.1, color = :red, label = "MPT", ylims = (-10,10),
        xlabel = "Time [kyrs BP]", xaxis = :on)

# d18O / GSL
po1 = plot(plot_LR04, xlabel = "", xaxis = :off, grid = true, xtickfont = false)
po2 = plot(plot_SL, xlabel = "", xaxis = :off, grid = true, xtickfont = false)
po3 = plot(plot_E, xlabel = "", xaxis = :off, grid = true, xtickfont = false)
po4 = plot(plot_R, xlabel = "", xaxis = :off, grid = true, xtickfont = false)
po5 = plot(plot_G, xlabel = "", xaxis = :off, grid = true, xtickfont = false)

# CO2
#po6 = plot(plot_pCO2, xlabel = "", xaxis = :off)
po6 = plot(plot_B, xlabel = "", xaxis = :off, grid = true, xtickfont = false)
po6_edc = plot(plot_B_edc, xlabel = "", xaxis = :off, grid = true, xtickfont = f
alse)
po7 = plot(plot_C, xlabel = "", xaxis = :off, grid = true, xtickfont = false)

# dust
po8 = plot(plot_L, xlabel = "", xaxis = :off, grid = true, xtickfont = false)
po8_edc = plot(plot_L_edc, xlabel = "", xaxis = :off, grid = true, xtickfont = f
alse)
po9 = plot(plot_MG, xlabel = "", xaxis = :off, grid = true, xtickfont = false,)
po10 = plot(plot_La2004, xlabel = "", xaxis = :off, grid = true, xtickfont = fal
se)

p_xaxis = plot(xlabel = "Time [kyrs BP]", xaxis = :on, xtickfont = 8, yaxis = :o
ff,)

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

#po_alltimeseries =
plot(po1, po2, po3, po4, po5, po7, po6, po6_edc, po8, po8_edc, po9, po10, p_xaxi
s, #p_highlight_MPT,
    size = (72*8.27, 72*11.69), # A4 (72 dots/inch)
    layout = l,
    xaxis = :on, xlims = (-5500,400), xticks = (-5000:250:0),
    yaxis = :off, ylabel = "",
    bg_legend = :transparent, #legend = :left,
    grid = :off,
    #Shape([-1250,-1250,-700,-700],[-100,100,100,-100]), fillalpha = 0.1, color
 = :red, label = "MPT",         # DOESN'T WORK
    )

#savefig("../../figurar/BinnedTimeseries/p_alltimeseries_2000_Appendix.pdf")
```

```
UndefVarError: Shape not defined

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

End of Notebook 1

Next steps:

Now that the datasets are binned to a regular time grid, we only need to select a common time interval between records to run our analyses. This is done in the NBRs.

Outline for our next notebooks:

NB2) In notebook 2, we thoroughly go through all the steps in our analysis. We use the example of a synthetic autoregressive system where we already know the causal coupling, and how the predictive asymmetry is estimated between timeseries data. This notebook shows all the steps in the analysis with abundant comments and explanatory figures, and aims to give a general understanding of what is done in the following notebooks (NB3 and NBRs).

(After synthetic time series are represented, include note on reflections/challenges for using real-world empirical time series.)

NB3) In notebook 3, is the *toolbox* we prepare to run our analysis. The wrangled data that we have prepared in this notebook (NB1) is read in to easily be included in the next notebooks (NBRs). We also write a function that synthesize the code for computing the predictive asymmetry (detailed in NB2). This function allows us to run the analysis between the many time series more efficiently.

NBRs) The results notebooks (NBR) is where we run the analyses between our time series data and plot our results. By including NB3 in the beginning of the notebook, we include the wrangled time series data (prepared in this notebook NB1) and the functions to run the analyses. We make a separate NBR for every grid (time interval) we will run analysis over. Each time grid is constrained primarily by the resolution and time span of the available records. (ice core records, for example, only cover the last 800 kyrs). We also define time span constraints (*windows*) on time grids that allow us to check if dynamical coupling differs over the hypothesized periods post-MPT (after 700 kyrs BP), syn-MPT (700-1250 kyrs BP) and pre-MPT (before 1.25 Myrs BP). The NBRs are named after the time grid they run analyses over. The grids we will use are:

- post-MPT
  - `postMPT_500` [-492:-12] (ca 0-500 kyrs BP, analysis delimited by the Grant record) - DONE
  - maybe `postMPT_700` (ca. 0-700 kyrs BP)
- syn-MPT
  - `synMPT` [-1250:-1080] (1.080-1.250 Myrs BP, analysis delimited by the Chalk record) - DONE
- pre_MPT
  - `preMPT_1500` [-1500:1:-1250] (1.250-1.500 Myrs BP, delimited by the Elderfield record) - DONE
  - maybe `preMPT_4000` [-4000:1:-1250] (1.250-4.000 Myrs BP, Rohling record only available proxy for gsl)
- (full grids)

  ```
  > What would we want to say with these?
  ```

  - `full_800` (0-800 kyrs BP) - almost done
    - all records (except Chalk)

    > If we were to prioritize between postMPT_700 and full_800 (same notebook), what would be more useful results?

  - `full_1500` (0-1500 kyrs BP)
    - Elderfield GSL, Rohling GSL, La2004 insolation, Martinez-García Fe dust.

- maybe `full_4000` (0-4000 kyrs BP)
  - Rohling GSL, La2004 insolation, Martinez-García Fe dust.

> is this interesting? Given the Rohling record is so-so. Maybe if we can cut out and interpolate over the sapropelic intervals, and the method still has statistical power. If not, I think better to leave it be.

In [ ]:

```

```

In [ ]:

```

```