Supplemental material to ’Eccentricity-paced atmospheric carbon-dioxide variations across the middle Miocene climate transition’

by Markus Raitzsch et al.

Figure S1. (A, B) Stable isotope records of IODP Site U1338 and ODP Site 1092. (C) Sea-surface temperatures based on *G. bulloides* Mg/Ca (Kuhnert et al., 2009), and corrected for seawater Mg/Ca following Evans and Müller (2012). The shaded area is the ±1 °C uncertainty. (D) Calculated pH of the sea surface (red) and the deep sea (green) derived from boron isotope measurements. The shaded areas represent 2σ uncertainties.

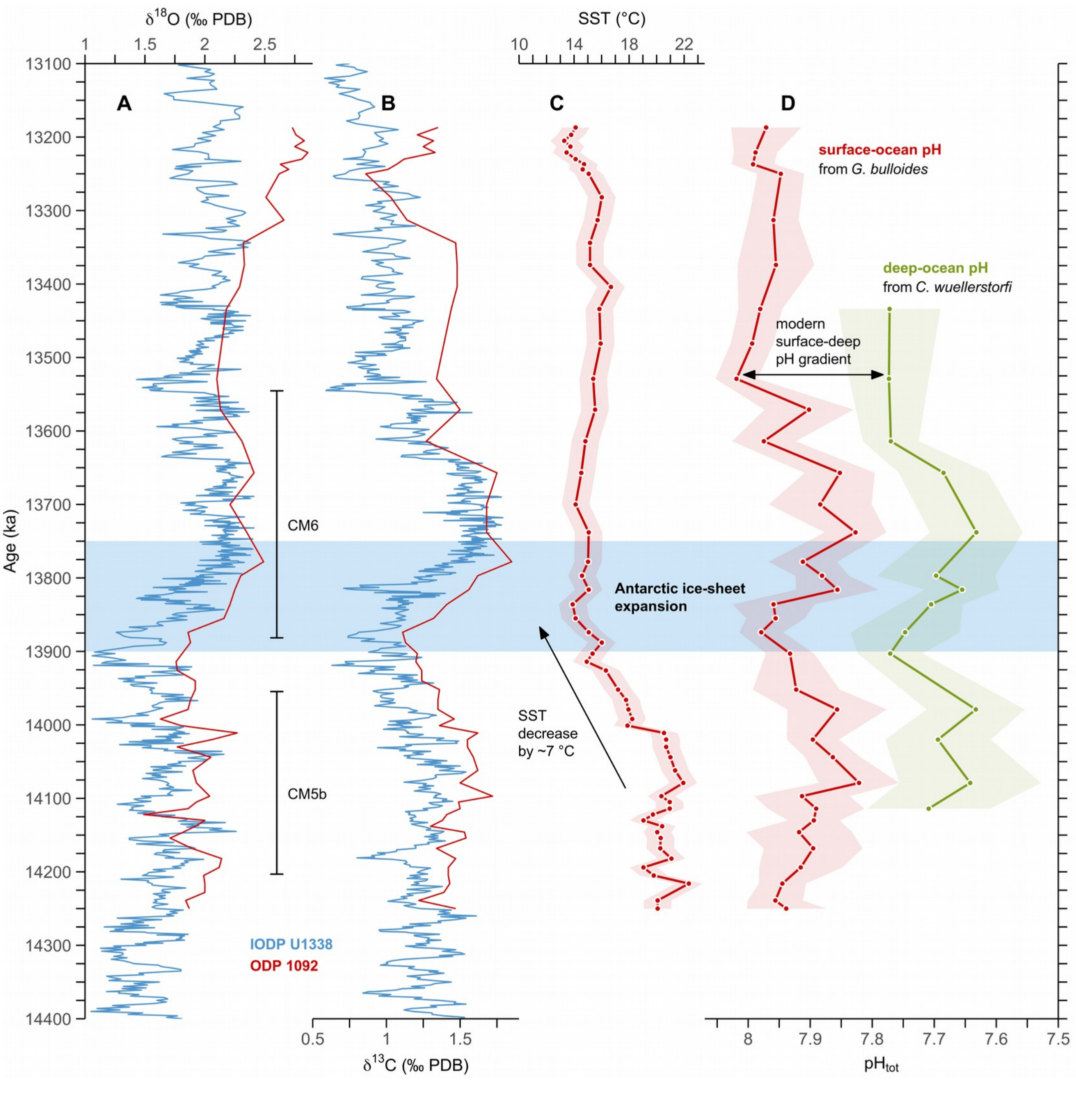

Figure S2. Upper panel: Sea-surface temperature reconstruction from ODP Site 1092, based on original Mg/Ca data (red), and based on iteratively pH-corrected Mg/Ca data (green), using the method by Gray and Evans (2019). Shaded area is propagated 1σ uncertainty. Lower panel: *p*CO$_2$ reconstruction based on conventionally δ^{11}B-derived pH data (red), and based on iteratively temperature-corrected pH data (green). Shaded area is 2σ uncertainty. Note that the two datasets are tied at the beginning of the interval to depict the relative differences between the two methods (see text for details). The grey line represents the box model output from Ma et al. (2011).

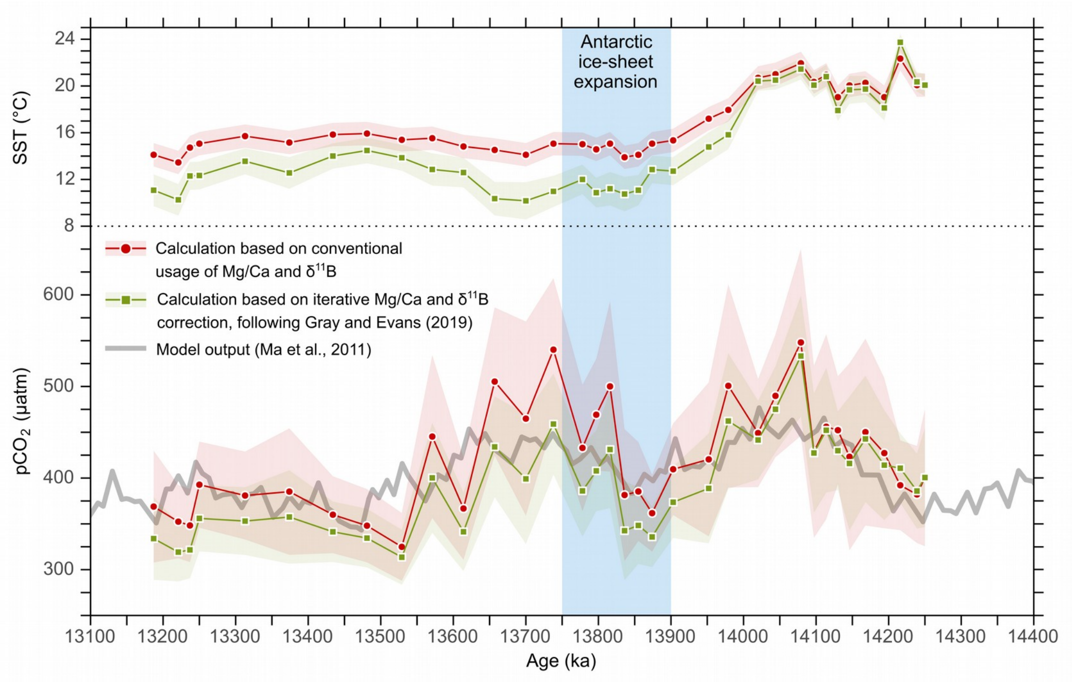

Table S1. Sampled core sections, boron isotope values, and reconstructed environmental parameters.

Globigerina bulloides										
Core section	Interval (cm)	Age (ka BP)	[1]Mg/Ca-T (°C)	[2]Salinity	$\delta^{11}B_{Cc}$ (‰)	[3]2σ $\delta^{11}B_{Cc}$	pH (tot)	2σ pH	*p*CO_2 (µatm)	[4]2σ *p*CO_2
177-1092B-18-2	130-132	13187	14.1	34.6	12.45	0.47	8.021	0.056	369	61
177-1092B-18-3	20-22	13221	13.5	34.5	12.53	0.30	8.038	0.035	352	40
177-1092B-18-3	40-42	13237	14.7	34.8	12.73	0.30	8.042	0.034	348	40
177-1092B-18-3	60-62	13250	15.1	34.8	12.35	0.30	7.997	0.038	393	47
177-1092B-18-3	80-82	13313	15.7	34.7	12.53	0.33	8.009	0.040	381	48
177-1092B-18-3	100-102	13374	15.2	34.6	12.42	0.50	8.005	0.062	385	69
177-1092B-18-3	120-122	13434	15.8	34.6	12.74	0.30	8.031	0.035	360	42
177-1092B-18-3	140-142	13481	15.9	34.4	12.87	0.30	8.044	0.034	348	40
177-1092B-18-4	0-2	13529	15.4	34.4	13.06	0.31	8.069	0.034	325	37
177-1092B-18-4	9-11	13571	15.5	34.2	11.97	0.52	7.952	0.072	445	89
177-1092B-18-4	19-21	13614	14.8	34.1	12.54	0.43	8.025	0.051	367	56
177-1092B-18-4	29-31	13657	14.5	34.5	11.48	0.37	7.902	0.056	505	81
177-1092B-18-4	39-41	13700	14.1	34.5	11.69	0.58	7.934	0.083	465	106
177-1092B-18-4	49-51	13738	15.1	34.5	11.34	0.31	7.877	0.050	540	78
177-1092B-18-4	59-61	13778	15.0	34.4	12.01	0.41	7.962	0.055	433	69
177-1092B-18-4	69-71	13797	14.1	34.4	11.71	0.30	7.931	0.043	469	61
177-1092B-18-4	79-81	13816	15.1	34.7	11.57	0.44	7.906	0.066	500	93
177-1092B-18-4	89-91	13836	13.9	34.3	12.30	0.54	8.009	0.066	382	72
177-1092B-18-4	99-101	13855	14.1	34.2	12.29	0.38	8.006	0.047	385	54
177-1092B-18-4	109-111	13874	15.1	34.5	12.63	0.30	8.029	0.035	362	42
177-1092B-18-4	131-133	13903	15.3	34.5	12.23	0.30	7.983	0.039	410	50
177-1092B-18-5	19-21	13952	17.2	34.8	12.36	0.53	7.973	0.070	420	84
177-1092B-18-5	39-41	13979	18.0	35.0	11.87	0.53	7.907	0.080	501	111
177-1092B-18-5	79-81	14020	20.7	35.8	12.55	0.30	7.946	0.042	449	59
177-1092B-18-5	99-101	14044	21.0	35.9	12.29	0.30	7.913	0.045	490	67
177-1092B-18-5	119-121	14079	21.9	36.0	12.02	0.40	7.871	0.065	548	102
177-1092B-18-5	139-141	14097	20.3	35.7	12.67	0.57	7.963	0.077	429	94
177-1092B-18-6	9-11	14114	21.0	35.7	12.52	0.55	7.940	0.078	456	101
177-1092B-18-6	29-31	14130	19.0	35.4	12.33	0.30	7.944	0.042	452	59
177-1092B-18-6	49-51	14146	20.0	35.7	12.68	0.64	7.968	0.085	423	102
177-1092B-18-6	69-71	14168	20.3	35.7	12.49	0.57	7.945	0.080	450	102
177-1092B-18-6	89-91	14194	19.0	35.4	12.53	0.50	7.965	0.067	427	82
177-1092B-18-6	109-111	14216	22.3	35.8	13.23	0.30	7.995	0.038	392	49
177-1092B-18-6	129-131	14239	20.1	35.4	13.05	0.36	8.006	0.044	382	53
177-1092B-18-6	139-141	14250	20.1	35.5	12.88	0.50	7.989	0.064	401	75

Table S1. continued

Cibicidoides wuellerstorfi										
Core section	**Interval (cm)**	**Age (ka BP)**	**[5]Mg/Ca-T (°C)**	**Salinity**	**$\delta^{11}B_{Cc}$ (‰)**	**[6]2σ $\delta^{11}B_{Cc}$**	**pH (tot)**	**2σ pH**	**pCO_2 (µatm)**	**2σ pCO_2**
177-1092B-18-3	120-122	13434	5.7	34.0	13.18	0.46	7.822	0.082	–	–
177-1092B-18-4	0-2	13529	5.0	34.0	13.13	0.36	7.823	0.064	–	–
177-1092B-18-4	19-21	13614	4.5	34.0	13.08	0.25	7.820	0.045	–	–
177-1092B-18-4	29-31	13657	4.5	34.0	12.63	0.34	7.735	0.072	–	–
177-1092B-18-4	49-51	13738	5.0	34.0	12.41	0.32	7.682	0.075	–	–
177-1092B-18-4	69-71	13797	5.6	34.0	12.76	0.46	7.747	0.095	–	–
177-1092B-18-4	79-81	13816	5.9	34.0	12.57	0.27	7.705	0.060	–	–
177-1092B-18-4	89-91	13836	5.8	34.0	12.81	0.25	7.755	0.051	–	–
177-1092B-18-4	109-111	13874	5.6	34.0	13.03	0.47	7.797	0.088	–	–
177-1092B-18-4	131-133	13903	5.0	34.0	13.12	0.29	7.821	0.052	–	–
177-1092B-18-5	39-41	13979	4.5	34.0	12.39	0.34	7.683	0.079	–	–
177-1092B-18-5	79-81	14020	4.6	34.0	12.68	0.41	7.744	0.085	–	–
177-1092B-18-5	119-121	14079	4.7	34.0	12.44	0.50	7.692	0.114	–	–
177-1092B-18-6	9-11	14114	5.4	34.0	12.81	0.49	7.759	0.099	–	–

[1] Sea-surface temperatures based on Mg/Ca ratios in *G. bulloides* from Kuhnert et al. (2009), and corrected for seawater Mg/Ca following Evans and Müller (2012).

[2] Relative salinity changes were estimated by converting $\delta^{18}O_{sw}$, which was derived from planktonic foraminiferal $\delta^{18}O$ and Mg/Ca temperatures, to salinity using a $\delta^{18}O_{sw}$:salinity gradient of 1.1 ‰ (Kuhnert et al., 2009). An offset of 17.2 (psu) was added to the entire salinity record to achieve post-glaciation values similar to today (see Section 3 for further details).

[3] The measurement uncertainty is based on 2 standard deviations from triplicate measurements, or the long-term uncertainty of a control standard (±0.30 ‰, n=48), whichever is larger.

[4] Uncertainties of pCO_2 estimates were fully propagated from individual uncertainties in pH (converted from 2σ measurement uncertainty of $\delta^{11}B$), TA (±100 µmol/kg), temperature (±1 °C), and salinity (±1 psu).

[5] Deep-sea temperatures based on Mg/Ca ratios in *C. wuellerstorfi* using the species-specific equation Mg/Ca=0.830*exp(0.145*T) (Raitzsch et al., 2008).

[6] The measurement uncertainty is based on 2 standard deviations from triplicate measurements, or the long-term uncertainty of a control standard (±0.25 ‰, n=12), whichever is larger.

Table S2. Tie points for revised age models

Site	Position	Original age (ka)	Revised age (ka)	Reference
ODP 1092	mcd	Kuhnert et al. (2009)	This study	tuned to
	178.92	13259	13252	IODP Site U1338
	180.21	13679	13698	IODP Site U1338
	180.41	13758	13778	IODP Site U1338
	180.91	13918	13875	IODP Site U1338
	181.81	14190	13991	IODP Site U1338
	182.21	14239	14034	IODP Site U1338
	182.51	14263	14081	top C5ADn
	183.41	14343	14155	IODP Site U1338
	183.61	14351	14183	IODP Site U1338
	184.11	14371	14239	IODP Site U1338
ODP 761	mcd	rev. by Sosdian et al. (2018)	This study	tuned to
	40.3	NA	13325	IODP Site U1338
	40.8	13385	13486	IODP Site U1338
	41.7	13763	13725	IODP Site U1338
	42.2	NA	13879	IODP Site U1338
	42.8	13937	13971	IODP Site U1338
	43.8	NA	14055	IODP Site U1338
	44.1	NA	14153	IODP Site U1338
	44.3	NA	14182	IODP Site U1338
Ras-il-Pellegrin A	Section height (m)	Abels et al. (2005)	This study	tuned to
	28.66	13096	13142	IODP Site U1338
	25.46	13182	13232	IODP Site U1338
	23.9	13234	13272	IODP Site U1338
	18.55	13360	13348	IODP Site U1338
	12.87	13530	13432	IODP Site U1338
	8.65	13635	13533	IODP Site U1338
	6.19	13696	13634	IODP Site U1338
	1.5	13825	13814	IODP Site U1338
	0.9	13840	13841	IODP Site U1338
	– 0.67	13880	13880	IODP Site U1338
	– 1.55	13902	13907	IODP Site U1338
	– 4.75	13983	13991	IODP Site U1338
Ras-il-Pellegrin B	Section height (m)	Badger et al. (2013)	This study	tuned to
	NA	13283	13104	revised Ras-il-PellegrinA
	NA	13339	13181	revised Ras-il-Pellegrin A
	NA	13462	13330	revised Ras-il-Pellegrin A
	NA	13535	13454	revised Ras-il-Pellegrin A
	11.2	13571	13503	revised Ras-il-Pellegrin A
	9.1	13628	13543	revised Ras-il-Pellegrin A
	8.4	13644	13569	revised Ras-il-Pellegrin A
	6.65	13679	13630	revised Ras-il-Pellegrin A
	5.25	13704	13701	revised Ras-il-Pellegrin A
	3.85	13731	13756	revised Ras-il-Pellegrin A
	0.7	13786	13838	revised Ras-il-Pellegrin A

S1. Modified ‘MgCaRB’ R code from Gray and Evans (2019) in detail (changes marked in bold red)**:**

```r
#MaCaRBv1 d11B function
MgCaRB.d11B <-
function(species,
           age,
           mgca,
           mgca_err,
           d11BOH4,
           d11BOH4_err,
           S,
           S_err) {
require(seacarb)
# The following part is deactivated to prevent error message at ages >798 ka
# if(min(age) < -0.05 | max(age) > 798) {print('check age range/units!')} else
# if(mean(age_err) > 99) {print('check age error units!')} else {
# The following part is deactivated, as the record is older than 798 ka
# #import S data
# esl_dat<- read.csv(paste(getwd(),'/spratt2016_esl.csv', sep='')) #use sprat2016
# esl_dat<- rbind(c(-0.05, esl_dat$esl[1]), esl_dat)
# Desl_total<- min(esl_dat$esl)-max(esl_dat$esl)
# esl_dat$DS_sl<-  esl_dat$esl*(0.7/Desl_total) #0.7 is mean surface change in S from model

#settings
    pHambient <-8.1
    pH_threshold <-0.0001
    T_threshold <-0.001

#boron constants
    d11B_sw <-37.8# This is changed to the Miocene value
    alpha_b <-1.0272
    epsilon_b <-1000*(alpha_b -1)

#mgca constants
#tsens, ssens, pHsens
    generic <-c(0.061249, 0.036136,-0.73150)
    grbw_gray2018 <-c(0.059759, 0.03313,-0.83263)
    grbw_spc <-c(0.06388, 0.03538,-0.87005)
    tsac_spc <-c(0.062413, 0.053976, 0)
    gbul_spc <-c(0.06411, 0.032966,-0.87527)
```

```
    ouni_spc <-c(0.07461, 0.04004, -0.49577)
    grbw <-c(0.061249, 0.036136,-0.87309)
    tsac <-c(0.061249, 0.036136, 0)
    gbul <-c(0.061249, 0.036136,-0.87738)
    ouni <-c(0.061249, 0.036136,-0.50927)

#mgca constants error
#tsens_err, ssens_err, pHsens_err
    generic_err <-c(0.005239, 0.006176, 0.07245)
    grbw_gray2018_err <-c(0.003763642, 0.011180065, 0.16933)
    grbw_spc_err <-c(0.01785, 0.01318, 0.11844)
    tsac_spc_err <-c(0.017841, 0.009464, 0.084416)
    gbul_spc_err <-c(0.03110, 0.006176, 0.13025)
    ouni_spc_err <-c(0.01210, 0.01427, 0.12053)
    grbw_err <-c(0.005239, 0.006176,  0.10969)
    tsac_err <-c(0.005239, 0.006176,  0.084416)
    gbul_err <-c(0.005239, 0.006176,  0.12103)
    ouni_err <-c(0.005239, 0.006176, 0.11786)

#mcmc loop
    mcmc_iterations <-999
    r1 <-matrix(, nrow =length(mgca), ncol = mcmc_iterations)
    r2 <-matrix(, nrow =length(mgca), ncol = mcmc_iterations)

for (j in1:mcmc_iterations) {
tryCatch({
        tsens <-rnorm(
1,
if (species =='generic') {
            generic[1]
          } else
if (species =='grbw_gray2018') {
              grbw_gray2018[1]
            } else
if (species =='grbw') {
                grbw[1]
              } else
if (species =='tsac') {
                  tsac[1]
```

```
                } else
if (species =='gbul') {
                    gbul[1]
                  } else
if (species =='ouni') {
                      ouni[1]
                    } else
if (species =='grbw_spc') {
                        grbw_spc[1]
                      } else
if (species =='tsac_spc') {
                          tsac_spc[1]
                        } else
if (species =='gbul_spc') {
                            gbul_spc[1]
                          } else
if (species =='ouni_spc') {
                              ouni_spc[1]
                            }
          ,
if (species =='generic') {
            generic_err[1]
          } else
if (species =='grbw_gray2018') {
              grbw_gray2018_err[1]
            } else
if (species =='grbw') {
                grbw_err[1]
              } else
if (species =='tsac') {
                  tsac_err[1]
                } else
if (species =='gbul') {
                    gbul_err[1]
                  } else
if (species =='ouni') {
                      ouni_err[1]
                    } else
if (species =='grbw_spc') {
```

```
                        grbw_spc_err[1]
                      } else
if (species =='tsac_spc') {
                          tsac_spc_err[1]
                        } else
if (species =='gbul_spc') {
                            gbul_spc_err[1]
                          } else
if (species =='ouni_spc') {
                              ouni_spc_err[1]
                            }
        )

        ssens <-rnorm(
1,
if (species =='generic') {
            generic[2]
          } else
if (species =='grbw_gray2018') {
              grbw_gray2018[2]
            } else
if (species =='grbw') {
                grbw[2]
              } else
if (species =='tsac') {
                  tsac[2]
                } else
if (species =='gbul') {
                    gbul[2]
                  } else
if (species =='ouni') {
                      ouni[2]
                    } else
if (species =='grbw_spc') {
                        grbw_spc[2]
                      } else
if (species =='tsac_spc') {
                          tsac_spc[2]
                        } else
```

```
if (species =='gbul_spc') {
                            gbul_spc[2]
                          } else
if (species =='ouni_spc') {
                              ouni_spc[2]
                            }
          ,
if (species =='generic') {
            generic_err[2]
          } else
if (species =='grbw_gray2018') {
              grbw_gray2018_err[2]
            } else
if (species =='grbw') {
                grbw_err[2]
              } else
if (species =='tsac') {
                  tsac_err[2]
                } else
if (species =='gbul') {
                    gbul_err[2]
                  } else
if (species =='ouni') {
                      ouni_err[2]
                    } else
if (species =='grbw_spc') {
                        grbw_spc_err[2]
                      } else
if (species =='tsac_spc') {
                          tsac_spc_err[2]
                        } else
if (species =='gbul_spc') {
                            gbul_spc_err[2]
                          } else
if (species =='ouni_spc') {
                              ouni_spc_err[2]
                            }
        )
```

```
    pHsens <-rnorm(
1,
if (species =='generic') {
      generic[3]
     } else
if (species =='grbw_gray2018') {
       grbw_gray2018[3]
      } else
if (species =='grbw') {
        grbw[3]
       } else
if (species =='tsac') {
         tsac[3]
        } else
if (species =='gbul') {
          gbul[3]
         } else
if (species =='ouni') {
           ouni[3]
          } else
if (species =='grbw_spc') {
            grbw_spc[3]
           } else
if (species =='tsac_spc') {
             tsac_spc[3]
            } else
if (species =='gbul_spc') {
              gbul_spc[3]
             } else
if (species =='ouni_spc') {
               ouni_spc[3]
              }
     ,
if (species =='generic') {
      generic_err[3]
     } else
if (species =='grbw_gray2018') {
       grbw_gray2018_err[3]
      } else
```

```
if (species =='grbw') {
                grbw_err[3]
              } else
if (species =='tsac') {
                  tsac_err[3]
                } else
if (species =='gbul') {
                    gbul_err[3]
                  } else
if (species =='ouni') {
                      ouni_err[3]
                    } else
if (species =='grbw_spc') {
                        grbw_spc_err[3]
                      } else
if (species =='tsac_spc') {
                          tsac_spc_err[3]
                        } else
if (species =='gbul_spc') {
                            gbul_spc_err[3]
                          } else
if (species =='ouni_spc') {
                              ouni_spc_err[3]
                            }
        )

#from fit error covariance
        intercept <-if (species =='generic') {
1.5+tsens *-25
        } else
if (species =='grbw_gray2018') {
rnorm(1, 1.83279+tsens *-29.18481, 0.0001141005)
          } else
if (species =='grbw_spc') {
rnorm(1, 1.563359+tsens *-25.016445, 0.0002490649)
            } else
if (species =='tsac_spc') {
rnorm(1, 1.365102+tsens *-25.704721, 0.0001442152)
              } else
```

```
if (species =='gbul_spc') {
rnorm(1, 1.564222+tsens *-22.053373, 0.0002023524)
                } else
if (species =='ouni_spc') {
rnorm(1, 2.091911+tsens *-21.679394, 0.0001874262)
                  } else
if (species =='grbw') {
rnorm(1, 1.623660+tsens *-25.960728, 0.000197117)
                    } else
if (species =='tsac') {
rnorm(1, 1.39803+tsens *-25.90757, 0.0001313673)
                      } else
if (species =='gbul') {
rnorm(1, 1.525943+tsens *-21.442429 , 0.0002160964)
                        } else
if (species =='ouni') {
rnorm(1, 2.136308+tsens *-22.297933, 0.000200698)
                          }

# The following code chunk is deactivated as specific salinity estimates are used
# #variable uncertainty
# agej<- rnorm(length(age), age, age_err)
# agej<- ifelse(agej >= -0.05, agej, -0.05) #min age limit for sea level/ice core data
# agej<- ifelse(agej <= 798, agej, 798) #max age limit for sea level data
#
# DS_esl<- approx(x=esl_dat$age, y=esl_dat$DS_sl, xout=agej)$y
# Sj<- rnorm(length(age), (S + DS_esl), 0.5) #S uncertainty from model

# Instead, the following line replaces the code chunk above
        Sj <-rnorm(length(S), S, S_err)

        mgcaj <-rnorm(length(mgca), mgca, mgca_err)
        d11BOH4j <-rnorm(length(d11BOH4), d11BOH4, d11BOH4_err)

#calculate T0
        T0 <-(1/tsens) *log(mgcaj /exp(ssens *(Sj -35) +pHsens *(pHambient -8) +intercept))
        i <-0

#iterative loop
```

```r
repeat {
#1
          i <-i +1
#calculate pH1
          pkb <--log(Kb(
S = Sj,
T = T0,
P =5,
warn ='n'
          ),
base =10)

          pH1 <-pkb -log(-((d11B_sw -d11BOH4j) /(d11B_sw -(alpha_b *d11BOH4j) -1000*(alpha_b
-1))), base =10)

#calculate T1
          T1 <-(1/tsens) *log(mgcaj /exp(ssens *(Sj -35) +pHsens *(pH1 -8) +intercept))


#2
          i <-i +1
#calculate pH2
          pkb <--log(Kb(
S = Sj,
T = T1,
P =5,
warn ='n'
          ),
base =10)

          pH2 <-pkb -log(-((d11B_sw -d11BOH4j) /(d11B_sw -(alpha_b *d11BOH4j) -1000*(alpha_b
-1))), base =10)

#calculate T2
          T2 <-(1/tsens) *log(mgcaj /exp(ssens *(Sj -35) +pHsens *(pH2 -8) +intercept))
          T0 <-T2

          pHdiff <-abs(pH2 -pH1)
          Tdiff <-abs(T2 -T1)
```

```r
ifelse(pHdiff <pH_threshold &
Tdiff <T_threshold, break, NA)
        }
#print(paste(i,'iterations to solve'))
        r1[, j] <-T2
        r2[, j] <-pH2
#s<- cbind(s,T2)
#s1<- cbind(s1,pH2)
#if (length(s[1,]) >= mcmc_iterations){break}
      },
error =function(e) {
      })
    }

    t <-apply(r1, 1, function(x) {
mean(x, na.rm =TRUE)
    })
    t_sigma <-apply(r1, 1, function(x) {
sd(x, na.rm =TRUE)
    })
    pH <-apply(r2, 1, function(x) {
mean(x, na.rm =TRUE)
    })
    pH_sigma <-apply(r2, 1, function(x) {
sd(x, na.rm =TRUE)
    })
    results <-data.frame(
t = t,
t_sigma = t_sigma,
pH = pH,
pH_sigma = pH_sigma
    )

return(results)
  }


save('MgCaRB.d11B', file ='MgCaRB.d11B.Rdata')
load('MgCaRB.d11B.Rdata')
```