



Supplement of

Atmospheric carbon dioxide variations across the middle Miocene climate transition

Markus Raitzsch et al.

Correspondence to: Markus Raitzsch (mraitzsch@marum.de)

The copyright of individual parts of the supplement might differ from the article licence.



Figure S1. Reconstructed *p*CO₂, calculated with different boron isotopic compositions of seawater. The shaded areas represent fully propagated uncertainty envelopes as described in section 1.4.5 in the main text. The blue vertical bar marks the EAIS expansion. The red line (calculated with $\delta^{11}B_{sw}=37.8 \pm 0.2$ ‰) is the reconstruction reported in Table S1 and discussed in the main text.



Figure S2. Reconstructed pCO₂, based on iteratively solved estimates of Mg/Ca-temperatures and δ^{11} B-based pH for (a) this study, (b) Foster et al. (2012), and (c) Badger et al. (2013). The preceding conversion of δ^{11} B_{Cc} to δ^{11} B_{borate} has a huge impact on reconstructed pH and hence pCO₂, depending on the choice of available calibrations. For *T. trilobus* (b, c), we used that from Foster et al. (2012), which was also originally applied in these studies. The blueish vertical bar marks the EAIS expansion.



Figure S3. Upper panel: Sea-surface temperature reconstruction from ODP Site 1092, based on unadjusted Mg/Ca data (green), and based on iteratively pH-corrected Mg/Ca data (red), using the method by Gray and Evans (2019). Lower panel: pCO_2 reconstruction based on unadjusted $\delta^{11}B$ -derived pH data (green), and based on iteratively temperature-corrected pH data (red). Shaded areas delimit 2σ uncertainties. The grey line represents the box model output from Ma et al. (2011).



Figure S4. (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), were corrected for seawater Mg/Ca (Evans and Müller, 2012) and pH (Gray and Evans, 2019). (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 S5. Sensitivity tests of pH-derived pCO_2 estimations. (a) Effect of varying salinity (derived from $\delta^{18}O$ changes) and TA (at modern correlation with S) on estimated pCO_2 . (b) Effect of TA on pCO_2 estimates, when TA is correlated (green) or anti-correlated (orange) with sea-surface temperature at a total range of 400 µmol/kg. (c-h) Effect of TA on pCO_2 estimates, when TA is (anti-)correlated with pH or eccentricity at maximum ranges of ±100 (orange) and ±400 µmol/kg (green). The blue bar delimits the time of Antarctic ice-sheet expansion. Shaded areas represent propagated 2σ envelopes. Note that the red line (based on constant TA of 2000 µmol/kg) is the record shown in Figs. 4e and 5.

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

				Globige	rina bulloide	S					
Core section	Interval (cm)	Age (ka BP)	Mg/Ca (mmol/mol)	¹ Mg/Ca-T (°C)	² Salinity	δ ¹¹ Bcc (‰)	$^{3}2\sigma \delta^{11}B_{Ce}$	⁴pH (tot)	⁵ 2σ pH	⁶ pCO ₂ (μatm)	⁷ 2σ <i>p</i> CO ₂
177-1092B-18-2	130-132	13187	1.77	10.11	34.6	12.45	0.47	8.076	0.089	318	79
177-1092B-18-3	20-22	13221	1.65	9.26	34.5	12.53	0.30	8.096	0.078	302	67
177-1092B-18-3	40-42	13237	1.89	11.25	34.8	12.73	0.30	8.090	0.074	307	66
177-1092B-18-3	60-62	13250	1.96	11.26	34.8	12.35	0.30	8.049	0.087	342	83
177-1092B-18-3	80-82	13313	2.10	12.52	34.7	12.53	0.33	8.053	0.081	339	78
177-1092B-18-3	100-102	13374	1.98	11.62	34.6	12.42	0.50	8.054	0.091	338	87
177-1092B-18-3	120-122	13434	2.13	13.03	34.6	12.74	0.30	8.070	0.071	325	67
177-1092B-18-3	140-142	13481	2.15	13.46	34.4	12.87	0.30	8.080	0.069	317	64
177-1092B-18-4	0-2	13529	2.03	12.90	34.4	13.06	0.31	8.106	0.059	295	53
177-1092B-18-4	9-11	13571	2.06	11.78	34.2	11.97	0.52	8.005	0.110	386	116
177-1092B-18-4	19-21	13614	1.91	11.50	34.1	12.54	0.43	8.069	0.080	326	74
177-1092B-18-4	29-31	13657	1.85	9.33	34.5	11.48	0.37	7.970	0.130	420	145
177-1092B-18-4	39-41	13700	1.77	9.15	34.5	11.69	0.58	8.005	0.133	383	138
177-1092B-18-4	49-51	13738	1.96	9.88	34.5	11.34	0.31	7.942	0.142	452	169
177-1092B-18-4	59-61	13778	1.95	10.92	34.4	12.01	0.41	8.016	0.111	374	113
177-1092B-18-4	69-71	13797	1.86	9.83	34.4	11.71	0.30	7.992	0.119	397	126
177-1092B-18-4	79-81	13816	1.96	10.21	34.7	11.57	0.44	7.971	0.131	419	147
177-1092B-18-4	89-91	13836	1.73	9.73	34.3	12.30	0.54	8.065	0.102	329	92
177-1092B-18-4	99-101	13855	1.77	10.11	34.2	12.29	0.38	8.061	0.097	332	89
177-1092B-18-4	109-111	13874	1.96	11.81	34.5	12.63	0.30	8.077	0.075	319	69
177-1092B-18-4	131-133	13903	2.02	11.67	34.5	12.23	0.30	8.032	0.088	359	88
177-1092B-18-5	19-21	13952	2.45	14.41	34.8	12.36	0.53	8.013	0.096	379	101
177-1092B-18-5	39-41	13979	2.67	14.78	35.0	11.87	0.53	7.948	0.115	449	141
177-1092B-18-5	79-81	14020	3.58	19.41	35.8	12.55	0.30	7.968	0.075	426	93
177-1092B-18-5	99-101	14044	3.70	19.46	35.9	12.29	0.30	7.938	0.091	461	119
177-1092B-18-5	119-121	14079	4.09	20.42	36.0	12.02	0.40	7.895	0.105	517	151
177-1092B-18-5	139-141	14097	3.45	19.07	35.7	12.67	0.57	7.983	0.086	409	100
177-1092B-18-6	9-11	14114	3.68	19.78	35.7	12.52	0.55	7.960	0.093	435	114
177-1092B-18-6	29-31	14130	3.00	16.82	35.4	12.33	0.30	7.974	0.082	419	98
177-1092B-18-6	49-51	14146	3.34	18.59	35.7	12.68	0.64	7.987	0.093	404	106
177-1092B-18-6	69-71	14168	3.42	18.69	35.7	12.49	0.57	7.968	0.092	426	111
177-1092B-18-6	89-91	14194	3.00	17.09	35.4	12.53	0.50	7.993	0.085	399	96
177-1092B-18-6	109-111	14216	4.27	22.71	35.8	13.23	0.30	7.996	0.056	394	69
177-1092B-18-6	129-131	14239	3.35	19.31	35.4	13.05	0.36	8.020	0.064	371	72
177-1092B-18-6	139-141	14250	3.35	19.01	35.5	12.88	0.50	8.006	0.074	384	84

Table S1. continued

Cibicidoides wuellerstorfi											
Core section	Interval (cm)	Age (ka BP)	Mg/Ca (mmol/mol)	⁸ Mg/Ca-T (°C)	Salinity	δ ¹¹ Bcc (‰)	⁹ 2σ δ ¹¹ B _{Ce}	¹⁰ pH (tot)	⁵ 2σ pH	pCO2 (µatm)	2σ <i>p</i> CO ₂
177-1092B-18-3	120-122	13434	1.88	6.96	34.0	13.18	0.46	7.871	0.085	-	-
177-1092B-18-4	0-2	13529	1.71	6.31	34.0	13.13	0.36	7.872	0.074	-	-
177-1092B-18-4	19-21	13614	1.55	5.65	34.0	13.08	0.25	7.874	0.066	-	-
177-1092B-18-4	29-31	13657	1.51	5.44	34.0	12.63	0.34	7.812	0.089	-	-
177-1092B-18-4	49-51	13738	1.72	6.35	34.0	12.41	0.32	7.766	0.095	-	-
177-1092B-18-4	69-71	13797	1.85	6.87	34.0	12.76	0.46	7.814	0.094	-	-
177-1092B-18-4	79-81	13816	1.85	6.85	34.0	12.57	0.27	7.785	0.080	-	-
177-1092B-18-4	89-91	13836	1.82	6.76	34.0	12.81	0.25	7.822	0.072	-	-
177-1092B-18-4	109-111	13874	1.76	6.50	34.0	13.03	0.47	7.856	0.086	-	-
177-1092B-18-4	131-133	13903	1.72	6.36	34.0	13.12	0.29	7.870	0.064	-	-
177-1092B-18-5	39-41	13979	1.69	6.23	34.0	12.39	0.34	7.765	0.097	-	-
177-1092B-18-5	79-81	14020	1.70	6.29	34.0	12.68	0.41	7.809	0.095	-	-
177-1092B-18-5	119-121	14079	1.81	6.69	34.0	12.44	0.50	7.767	0.109	_	_
177-1092B-18-6	9-11	14114	1.82	6.73	34.0	12.81	0.49	7.823	0.096	_	_

¹Sea-surface temperatures based on Mg/Ca ratios in *G. bulloides* from Kuhnert et al. (2009), and iteratively corrected for the pH effect on Mg/Ca, following Gray and Evans (2019).

²Relative salinity changes were taken from Kuhnert et al. (2009) who converted $\delta^{18}O_{sw}$, derived from planktonic foraminiferal $\delta^{18}O$ and Mg/Ca temperatures, to salinity using a $\delta^{18}O_{sw}$:salinity gradient of 1.1 ‰. An offset of 17.2 (psu) was added to the entire salinity record to achieve post-glaciation values similar to today.

³The measurement uncertainty is based on 2 standard deviations from triplicate measurements using SEM detectors, or the long-term uncertainty of a control standard ($\pm 0.30 \%$, n=48), whichever is larger.

⁴pH was iteratively corrected for Mg/Ca-based temperatures, using the method of Gray and Evans (2019). In addition, the effect of Miocene seawater [Mg, Ca] on the dissociation constants of carbonic and boric acids was taken into account, following Hain et al. (2015).

⁵propagated pH uncertainty was derived from the MgCaRB tool (see Gray and Evans, 2019), and additionally includes 2σ uncertainties in the $\delta^{11}B_{cc}/\delta^{11}B_{borate}$ calibration equation, and $\delta^{11}B_{sw}$ (0.20 ‰).

 ${}^{6}pCO_{2}$ calculated using ${}^{4}pH$, TA=2000 µmol/kg, ${}^{2}salinity$, ${}^{1}adjusted Mg/Ca temperatures$, and a water depth of 50 m. In addition, the effect of Miocene seawater [Mg, Ca] on the dissociation constants of carbonic acid was taken into account, following Hain et al. (2015).

⁷Uncertainties of pCO_2 estimates were propagated from uncertainties in ⁵pH, TA (±150 µmol/kg), temperature (±1 °C), and salinity (±1 unit).

⁸Deep-sea temperatures based on Mg/Ca ratios in *C. wuellerstorfi* using the species-specific equation Mg/Ca= $0.825 \times 0.830 \times \exp(0.145 \times T)$ (Raitzsch et al., 2008), where the pre-exponential constant was supplemented with a correction factor of 0.825 (in bold) to account for the Mg/Ca_{sw} in the Miocene (Evans and Müller, 2012).

⁹The measurement uncertainty is based on 2 standard deviations from triplicate measurements using Daly detectors, or the long-term uncertainty of a control standard ($\pm 0.25 \%$, n=12), whichever is larger.

¹⁰Deep-sea pH was calculated using a calibration $\delta^{11}B_{borate} = \delta^{11}B_{Cc}*1.16 (\pm 0.37) - 2.53 (\pm 5.83)$ through core-top data of *C. wuellerstorfi* from Rae et al. (2011) and Raitzsch et al. (2020). In addition, the effect of Miocene seawater [Mg, Ca] on the dissociation constants of carbonic and boric acids was taken into account, following Hain et al. (2015).

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-Pellegrin A
	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 *red bold italic*):

```
#MaCaRBv1 d11B function
MgCaRB.d11B <-
function(species,
           age,
           mgca,
           mgca err,
           d11B0H4,
           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</pre>
# esl dat<- rbind(c(-0.05, esl_dat$esl[1]), esl_dat)</pre>
# Desl total<- min(esl dat$esl)-max(esl dat$esl)</pre>
# esl dat$DS sl<- esl_dat$esl*(0.7/Desl_total) #0.7 is mean surface change in S from model</pre>
```

#settings

```
pHambient <-8.1
pH_threshold <-0.0001
T_threshold <-0.001</pre>
```

#boron constants

dllB_sw <-37.8 # This is changed to the Miocene value dllB_sw_err <-0.2 # quoted uncertainty for Miocene seawater dllB alpha_b <-1.0272 epsilon_b <-1000*(alpha_b -1)</pre>

#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)</pre>
```

gbul <-c(0.061249, 0.036136, -0.87738)</pre> 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)</pre>
    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)</pre>
    gbul spc err <-c(0.03110, 0.006176, 0.13025)
    ouni_spc_err <-c(0.01210, 0.01427, 0.12053)</pre>
    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</pre>
    r1 <-matrix(, nrow =length(mgca), ncol = mcmc iterations)</pre>
    r2 <-matrix(, nrow =length(mgca), ncol = mcmc_iterations)</pre>
for (j inl:mcmc_iterations) {
```

tryCatch({

```
tsens <-rnorm(</pre>
```

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[<mark>2</mark>]
                     } 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(</pre>
1,
if (species =='generic') {
            generic[3]
          } else
if (species =='grbw_gray2018') {
              grbw_gray2018[3]
            } else
if (species =='grbw') {
                 grbw[<mark>3</mark>]
              } else
if (species =='tsac') {
                  tsac[3]
                 } else
if (species =='gbul') {
                     gbul[<mark>3</mark>]
                   } 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') {</pre>
```

```
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 for Salinity replaces the code chunk above
```

```
Sj <-rnorm(length(S), S, S_err)</pre>
```

mgcaj <-rnorm(length(mgca), mgca, mgca_err)</pre>

```
# Account for effect of seawater Mg/Ca on shell Mg/Ca
        mqcaj <- rnorm(length(mgca), mgca*((2.5*5.3^0.41)/2.5*([Mg/Ca]sw^0.41)), mgca err)</pre>
        d11B0H4j <-rnorm(length(d11B0H4), d11B0H4, d11B0H4 err)</pre>
        d11Bswj <- rnorm(length(d11B0H4), d11B sw, d11B sw err) # Generate range of possible d11Bsw
#calculate T0
        T0 <-(1/tsens) *log(mgcaj /exp(ssens *(Sj -35) +pHsens *(pHambient -8) +intercept))
        i <-0
#iterative loop
repeat {
#1
          i <-i +1
#calculate pH1
          pkb <--log(Kb(</pre>
S = Si,
T = T0,
P =5,
warn ='n'
          ),
base =10)
          pH1 <-pkb -log(-((d11Bswj -d11B0H4j) /(d11Bswj -(alpha_b *d11B0H4j) -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(</pre>
S = Sj,
T = T1,
P =5.
warn ='n'
          ),
base =10)
          pH2 <-pkb -log(-((d11Bswj -d11B0H4j) /(d11Bswj -(alpha b *d11B0H4j) -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)</pre>
          Tdiff <-abs(T2 -T1)</pre>
ifelse(pHdiff <pH threshold &</pre>
Tdiff <T threshold, break, NA)</pre>
        }
#print(paste(i,'iterations to solve'))
        r1[, j] <-T2
        r2[, j] <-pH2
#s<- cbind(s,T2)</pre>
#s1<- cbind(s1,pH2)</pre>
#if (length(s[1,]) >= mcmc iterations){break}
      },
error =function(e) {
      })
    }
    t <-apply(r1, 1, function(x) {</pre>
mean(x, na.rm =TRUE)
    })
    t sigma <-apply(r1, 1, function(x) {</pre>
sd(x, na.rm =TRUE)
    })
    pH <-apply(r2, 1, function(x) {</pre>
mean(x, na.rm =TRUE)
    })
    pH_sigma <-apply(r2, 1, function(x) {</pre>
sd(x, na.rm =TRUE)
    })
    results <-data.frame(</pre>
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')
```