
NATIONAL CORAL REEF MONITORING PROGRAM

Standard Operating Protocol

for

Carbonate Chemistry - Seacarb (Climate)

Editors: Coral Reef Conservation Program, NCRMP Atlantic and Pacific
Climate Teams, Atlantic Oceanographic and Meteorological Laboratory, Pacific
Islands Fisheries Science Center



NOAA
CORAL REEF
CONSERVATION PROGRAM

National Coral Reef Monitoring Program

Carbonate Chemistry-Seacarb Procedure

Overview:

Seacarb is a seawater carbonate chemistry package, coded in R, and available from CRAN (the Comprehensive R Archive Network) at <https://cran.r-project.org/web/packages/seacarb/>. As of this writing (April/2023) its current release is 3.3.1. Members of AOML's Coral Program take water samples and analyze them for DIC, TA, and density. Those parameters are then input into the *Seacarb* functions (mainly *carb*, but *pHinsi* is useful as well), to calculate $p\text{CO}_2$, pH, Aragonite Saturation state, and many other parameters.

How to use *Seacarb*:

1. Generally, it is easiest to interact with R data and packages using RStudio, which is a GUI front-end to the R software environment. R is available from <https://www.r-project.org> and RStudio from <https://posit.co/products/open-source/rstudio/>. Coral Program team members may also choose to access a client-server RStudio installation on our internal *shinycoral* server.
2. *Seacarb* can be installed from within R with the command: `install.packages('seacarb')`. The `install.packages` function is part of the R *util* package.
3. Our use of the `Seacarb::carb` function requires five inputs taken from environmental carbonate chemistry data.
4. Two of these input parameters are measured *in situ* at the time of sampling: Temperature in °C and Pressure in bar. Pressure is equal to 0 at the surface, and for samples at depth we approximate as: $Pressure (bar) = Depth (m) / 10$.
5. Salinity is another input parameter, and the most accurate measure of Salinity (PSU) is calculated from Density and Temperature, where density is measured to a high degree of accuracy using a densitometer, and Temperature is the *in situ* Temperature measured at collection time as mentioned above. If densitometry results are not available then the *in situ* Salinity measured at the time of collection (e.g. using one of our handheld YSI Conductivity meters) may be substituted.
6. The remaining trio of possible input parameters consist of Dissolved Inorganic Carbon (DIC), Total Alkalinity (TA, also called ALK in the *Seacarb* documentation) and pH. Any two of these parameters may be used as inputs and the third will be calculated as one of the outputs. DIC and TA must be in mol/kg. Our database supplies these parameters in either $\mu\text{mol/L}$ or $\mu\text{mol/kg}$ (the conversion between the two involves sample density), so care must be taken to select the $\mu\text{mol/kg}$ formulations and then to divide by 10^6 (to convert from micromoles to moles). Also, whichever two parameters are chosen as inputs must be expressed in the context of *in situ* sample collection. This may require conversion from DIC, TA or pH values reported in analysis context (i.e., at the temperature during analysis and not the temperature at the time of collection).
7. `Seacarb::carb` uses generic *var1* and *var2* arguments for these inputs, which have different meanings depending on what *flag* argument is specified. The Coral Program primarily uses *flag* = 15 (where *var1* = TA and *var2* = DIC, and pH is among the output parameters) but we also make use of *flag* = 8 (*var1* = pH and *var2* = TA) and *flag* = 9 (*var1* = pH and *var2* = DIC).

8. These five input parameters (Temperature, Pressure, Salinity, and any two of DIC, TA and pH) are created as R vectors. It is beyond the scope of this discussion to explore all the ways of loading data into R vectors, but common practices may include cutting and pasting, reading directly from CSV or Excel files, or querying from our MySQL database.
9. All other `Seacarb::carb` parameters are left as their package defaults. Check the current package documentation for exact details, but as of this writing, these include familiar parameters such as:
 - a. $k1k2 = x$ (using K1 and K2 from Lueker et al. (2000) assuming T is within the range 2 to 35 °C and S is within the range 19 to 43)
 - b. $ks = d$ (Ks from Dickson (1990))
 - c. $pHscale = T$ (total scale)
 - d. $b = u74$ (concentration of total boron from Uppstrom (1974))
10. The following code snippet shows an example of a call to `Seacarb::carb`, where the `bottle.data` data frame was pre-loaded by querying one of our MySQL views:

```

library(seacarb)

# Run seacarb
seacarb.output = carb(

  # flag refers to which two CO2 system variables are
  # being used; here we are using ALK and DIC
  flag = 15,

  # need to convert TA from umol/kg to mol/kg
  var1 = bottle.data$calc_avg_ta_kg_corrected / 10 ^ 6,

  # need to convert DIC from umol/kg to mol/kg
  var2 = bottle.data$calc_avg_dic_kg_corrected / 10 ^ 6,

  # salinity, in psu
  S = bottle.data$calc_salinity,

  # temperature, in deg C
  T = bottle.data$collected_temp,

  # pressure, need to convert from dbar to bar
  P = bottle.data$collected_depth / 10
)

```

11. The output from `Seacarb::carb` is a data frame of the same length as the input vectors, and includes columns for all of the input parameters as well as a number of useful calculated parameters, including *pH*, *pCO₂*, *OmegaAragonite* and *OmegaCalcite*.
12. It is our practice to check CRAN regularly for `Seacarb` code updates as well as keeping R and RStudio themselves current with their latest releases. We also explicitly note which version of `Seacarb` was used when doing our calculations, and we regularly resubmit past inputs to newer releases of `Seacarb` to check for different outcomes (keeping detailed histories if different calculated outcomes are detected). The last time we noted such differences in calculated outcomes was with the release of `Seacarb` 3.3.0 in October 2021 (see `Seacarb` changelog here: <https://github.com/cran/seacarb/blob/master/ChangeLog>).