An introduction to CHNOSZ

Jeffrey M. Dick

October 20, 2015

1 About

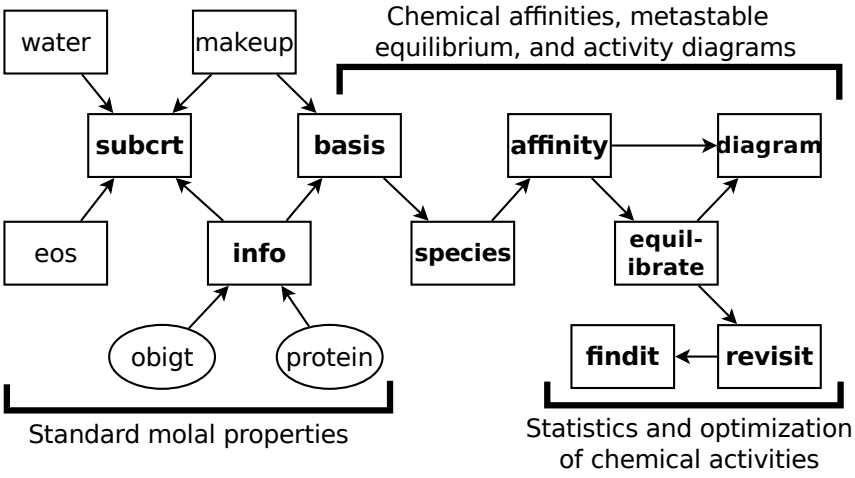
This document will orient you to the basic functionality of CHNOSZ, a package for the R software environment. R is a powerful language and also very fun to use. Don't worry if you're new to it; just plow through the examples below and you'll start to get the hang of it. If you want a more structured approach to learning the language, there are some excellent guides in the Manuals section of the R Project page.

The package was developed since 2006 to support a research project on the thermodynamic properties of proteins. Since that time, the functions in the package have expanded to include calculation of the thermodynamic properties of reactions, and especially the construction of equilibrium chemical activity diagrams for both inorganic and organic systems. The development of the package since 2009 has focused on the calculation of the equilibrium chemical activities of large numbers of proteins with applications to interpretation of metagenomic data and protein abundances in a variety of settings.

The database and functions are flexible in their use, allowing one to model the relative stabilities of proteins, minerals or aqueous species using very similar commands. Examples below are intended to demonstrate basic usage to new users.

1.1 Outline of workflow

CHNOSZ is made up of a set of functions and supporting datasets. The major components of the package are shown in the figure below, which is an updated version of the flowchart from Dick (2008) (boxes – functions; ellipses – datasets; bold text – major user functions).



Some common usage scenarios are:

- using `info()` to search for species in the thermodynamic database;
- using `subcrt()` to calculate the thermodynamic properties of species and reactions;

- using the sequence `basis()`, `species()`, `affinity()`, `equilibrate()`, `diagram()` to assign the basis species that define the dimensions of chemical composition in a system, define the species of interest for relative stability calculations, calculate the affinities of formation reactions of the species of interest under reference (non-equilibrium) conditions, calculate the equilibrium chemical activities, and finally plot the results;[1]
- using `revisit()` to calculate/plot statistics of the chemical activities of the species of interest and `findit()` to search for combinations of activities of basis species, temperature and/or pressure that optimize those statistics. These features, first appearing in version 0.9-3 of the package, are covered briefly toward the end of this document.

The functions are designed with an interactive setting in mind; you can use CHNOSZ without having to write your own scripts. The examples in this vignette are meant to portray a simple interactive session. However, as you become more familiar with CHNOSZ and R, you will probably find it helpful to save sequences of function calls that produce interesting results. The results can then be reproduced on demand by yourself or others with whom you might share your scripts.

1.2 Installing and loading CHNOSZ

If you have just installed R, and you are online, installing the CHNOSZ package should be as simple as selecting "Install packages from CRAN" or similar menu item in the R GUI or using the following command to start the package installation process. (If you are not online, you instead have to tell R to install the package from a local package file.)

```
> install.packages("CHNOSZ")
```

Then load the CHNOSZ package to make its functions available in your working session.

```
> library(CHNOSZ)
```

Then load the `thermo` object, which contains the thermodynamic database and is also where your system settings will be stored.

```
> data(thermo)
```

The rest of this document assumes that the CHNOSZ package and data are loaded.

2 Thermodynamic database

2.1 Warning about internal consistency of thermodynamic data

All thermodynamic data and examples are provided on an as-is basis. It is up to you to check not only the accuracy of the data, but also *the suitability of the data AND computational techniques* for your problem. By combining data taken from different sources, it is possible to build an inconsistent and/or nonsensical calculation. An attempt has been made to provide a primary database (OBIGT.csv) that is internally consistent, but no guarantee can be made. Where possible, data with known or suspected inconsistencies have been placed into a secondary database (OBIGT-2.csv) that should be regarded as experimental. If there is any doubt about the accuracy or suitability of data for a particular problem, please *consult the primary sources* (which can be located using `browse.refs()`; see Section 2.3). Do not assume that by adding any species to your calculation (or to any of the examples), you will necessarily obtain a reasonable answer. Do not assume that the examples are correct, or that they can be applied to your problem. As with the data, please *compare the construction and output of the examples to the primary sources*, cited in the reference list in each help page. Examples without a reference (and some with references) demonstrate experimental features of CHNOSZ.

[1] `equilibrate()` appeared in version 0.9-9 of CHNOSZ. In previous versions, the equilibrium calculations were invoked by calls to `diagram()`.

2.2 `info()` part I

So you want to know what are the standard molal thermodynamic properties and equations of state parameters of aqueous ethylene? Look no further than the `info()` function, which provides a convenient interface to retrieve entries from the thermodynamic database packaged with CHNOSZ.

```
> info("ethylene")

info.character: found ethylene(aq), also available in gas
[1] 88
```

There are two species named "ethylene" in the database. Normally, `info()` gives preference to aqueous species if they exist, so in this case we find that aqueous ethylene is species number 88 in the database. Let's display this entry, now by giving the species index to the function.

```
> info(88)

       name abbrv formula state ref1 ref2     date     G    H    S   Cp    V     a1
88 ethylene  <NA>    C2H4    aq SH90 <NA> 4.Sep.87 19450 8570 28.7 62.5 45.5 0.7856
        a2      a3     a4   c1    c2  omega Z
88 1263.91 -1.8737 -33014 39.1 97000 -40000 0
```

If you were instead interested in the properties of the gas, you could run:

```
> info("ethylene","gas")

[1] 3102
```

`info()` itself is used by other functions in the package. It prints output to the screen, but also returns a numeric value if it finds a species matching the search term. So, we can retrieve the properties of aqueous acetic acid without having to type in the species ID number.

```
> aadata <- info(info("acetic acid"))

info.character: found acetic acid(aq), also available in liq

> print(aadata)

           name abbrv formula state  ref1 ref2     date      G       H    S    Cp     V
515 acetic acid  <NA>  C2H4O2    aq Sho95 <NA> 6.Mar.92 -94760 -116100 42.7 40.56 52.01
         a1    a2     a3     a4     c1     c2  omega Z
515 1.16198 521.8 2.5088 -29946 42.076 -15417 -15000 0
```

2.3 `thermo$refs`

The thermodynamic data and other parameters used by the functions, as well as system definitions provided by the user in an interactive session, are stored in a list object called `thermo`.

```
> summary(thermo)

        Length Class      Mode
opt     17     -none-     list
element  6     data.frame list
obigt   20     data.frame list
refs     5     data.frame list
buffers  4     data.frame list
protein 25     data.frame list
groups  22     data.frame list
basis    0     -none-     NULL
species  0     -none-     NULL
Psat     0     -none-     NULL
opar    66     -none-     list
```

Within this list, the thermodynamic database is contained in a data frame (an R object that is like a matrix with named columns), `thermo$obigt`, and the references to the original sources of thermodynamic data in the literature are listed in `thermo$refs`. Many of the authors who are responsible for these data would be grateful if you cite them whenever these data are used in publications! Use the `browse.refs()` function without any arguments to show citation information for all of the references in a browser window. You can include a species index number to open the URL(s) associated with that entry in the database (this requires an Internet connection).

```
> browse.refs(88)

   browse.refs:  opening URL for SH90 (E. L. Shock and H. C. Helgeson, 1990)
```

2.4 `info()` part II

Want to know what acids are in the database?

```
> info("acid")

info.approx: 'acid' is ambiguous; has approximate matches to 75 species (showing first 25)
 [1] "a-aminobutyric acid" "formic acid"         "acetic acid"
 [4] "propanoic acid"      "n-butanoic acid"     "n-pentanoic acid"
 [7] "n-hexanoic acid"     "n-heptanoic acid"    "n-octanoic acid"
[10] "n-nonanoic acid"     "n-decanoic acid"     "n-undecanoic acid"
[13] "n-dodecanoic acid"   "n-benzoic acid"      "o-toluic acid"
[16] "m-toluic acid"       "p-toluic acid"       "oxalic acid"
[19] "malonic acid"        "succinic acid"       "glutaric acid"
[22] "adipic acid"         "pimelic acid"        "suberic acid"
[25] "azelaic acid"
[1] NA
```

Here, `info()` couldn't find an exact match to a name, so it performed a fuzzy search. That's why "uracil" and "metacinnabar" show up above. If you really just want species whose names include the term "acid", you can add a placeholder character to narrow the search. (Note: don't use an underscore ("_") here because that character is reserved for names of proteins. Any other character will do; here we use a space.)

```
> info(" acid")

info.approx: ' acid' is ambiguous; has approximate matches to 68 species (showing first 25)
 [1] "a-aminobutyric acid" "formic acid"         "acetic acid"
 [4] "propanoic acid"      "n-butanoic acid"     "n-pentanoic acid"
 [7] "n-hexanoic acid"     "n-heptanoic acid"    "n-octanoic acid"
[10] "n-nonanoic acid"     "n-decanoic acid"     "n-undecanoic acid"
[13] "n-dodecanoic acid"   "n-benzoic acid"      "o-toluic acid"
[16] "m-toluic acid"       "p-toluic acid"       "oxalic acid"
[19] "malonic acid"        "succinic acid"       "glutaric acid"
[22] "adipic acid"         "pimelic acid"        "suberic acid"
[25] "azelaic acid"
[1] NA
```

The names of species other than proteins use (almost) exclusively lowercase letters. `info()` can also be used to search the text of the chemical formulas as they are entered in the database; the symbols for the elements always start with a capital letter. The example below lists the formulas of aqueous species, then minerals, that contain the symbol commonly used to represent the hydroxide group.

```
> info("(OH)")
```

```
info.approx: '(OH)' is ambiguous; has approximate matches to 251 species (showing first 25)
 [1] "B(OH)3"                                  "U(OH)+3"
 [3] "Pd(OH)2"                                 "17a(H)-22,25,29,30-tetrakisnorhopane"
 [5] "17a(H)-22,29,30-trisnorhopane"           "17a(H)-29,30-bisnorhopane"
 [7] "17a(H)-30-norhopane"                     "17a(H)-hopane"
 [9] "17a(H)-homohopane"                       "17a(H)-bishomohopane"
[11] "17a(H)-trishomohopane"                   "17a(H)-tetrakishomohopane"
[13] "17a(H)-pentakishomohopane"               "17a(H)-hexakishomohopane"
[15] "17a(H)-heptakishomohopane"               "17a(H)-octakishomohopane"
[17] "17a(H)-nonakishomohopane"                "17a(H)-decakishomohopane"
[19] "17b(H)-22,25,29,30-tetrakisnorhopane" "17b(H)-22,29,30-trisnorhopane"
[21] "17b(H)-29,30-bisnorhopane"               "17b(H)-30-norhopane"
[23] "17b(H)-hopane"                           "17b(H)-homohopane"
[25] "17b(H)-bishomohopane"
[1] NA
```

3 Proteins

3.1 protein()

There are few things more fun than calculating the standard molal Gibbs energy of formation from the elements at 25 °C and 1 bar of a protein using group additivity. And there are few proteins whose thermodynamic properties are more well studied than lysozyme from the egg of the chicken.

```
> ip <- iprotein("LYSC_CHICK")
> aa <- ip2aa(ip)
> aa2eos(aa)

aa2eos: found LYSC_CHICK (C613H959N193O185S10, 129 residues)
        name abbrv             formula state   ref1 ref2 date        G         H       S
1 LYSC_CHICK    NA C613H959N193O185S10    aq BBA+03   NA   NA -4119738 -10283083 4176.74
        Cp        V    a1.a   a2.b   a3.c   a4.d   c1.e   c2.f omega.lambda z.T
1 6415.553 10420.89 2512.58 345.88 450.87 -409.5 7768.7 -701.5        -7.94   0
```

What happened there? Well, the first line found the row number (6) of `thermo$protein` that contains the amino acid composition of LYSC_CHICK. The second line extracted as a data frame. The third line used amino acid group contributions (Dick et al., 2006) to calculate the standard molal thermodynamic properties and equations of state parameters of the aqueous protein species. There are other functions available for calculating e.g. the chemical formula of the protein.

```
> pf <- protein.formula(aa)
> as.chemical.formula(pf)

[1] "C613H959N193O185S10"
```

3.2 info()

Most of the time you probably won't be using the `iprotein()` function. That's because `info()` recognizes the underscore character as being an essential part of the name of a protein. The names of proteins in CHNOSZ are mostly consistent with those used in Swiss-Prot/UniProtKB.

```
> si <- info("LYSC_CHICK")

aa2eos: found LYSC_CHICK (C613H959N193O185S10, 129 residues)

> info(si)
```

```
          name abbrv             formula state   ref1 ref2 date        G         H
3369 LYSC_CHICK  <NA> C613H959N193O185S10    aq BBA+03 <NA> <NA> -4119738 -10283083
           S       Cp        V      a1    a2     a3       a4     c1       c2   omega Z
3369 4176.74 6415.553 10420.89 251.258 34588 450.87 -4095000 7768.7 -7015000 -794000 0
```

When CHNOSZ is first loaded, the thermodynamic properties and parameters of the proteins are not present in `thermo$obigt`. Therefore, the first call to `info()` just above had a side effect of adding the computed properties and parameters to `thermo$obigt`.

4 Reaction properties

4.1 A single species

A major feature of CHNOSZ is the ability to calculate standard molal properties of species and reactions as a function of temperature and pressure. The function used is called `subcrt()`, which takes its name (with modification) from the well known SUPCRT package (Johnson et al., 1992). `subcrt()`, like `info()`, has the name of a species (including proteins) as its first argument (it also works if you give it the numeric index of the species in the database). If no reaction coefficients are given, the function calculates the standard molal properties of the indicated species on a default temperature-pressure grid.

```
> subcrt("water")

subcrt: 1 species at 15 values of T and P (wet)
$species
   name formula state ispecies
1 water     H2O   liq        1

$out
$out$water
        T          P       rho     logK         G         H        S        V       Cp
1    0.01   1.000000 0.9998289 45.03529 -56289.50 -68767.75 15.13238 18.01828 18.20559
2   25.00   1.000000 0.9970614 41.55247 -56687.71 -68316.76 16.71228 18.06830 18.01160
3   50.00   1.000000 0.9880295 38.63281 -57123.89 -67866.54 18.16234 18.23346 18.00464
4   75.00   1.000000 0.9748643 36.15435 -57594.93 -67416.13 19.50485 18.47970 18.04163
5  100.00   1.013220 0.9583926 34.02698 -58098.40 -66963.78 20.75956 18.79731 18.15793
6  125.00   2.320144 0.9390726 32.18315 -58631.71 -66507.34 21.94192 19.18403 18.33334
7  150.00   4.757169 0.9170577 30.57178 -59193.26 -66045.55 23.06398 19.64456 18.56643
8  175.00   8.918049 0.8923427 29.15313 -59781.38 -65576.63 24.13602 20.18866 18.88296
9  200.00  15.536499 0.8647434 27.89596 -60394.50 -65097.99 25.16818 20.83300 19.32884
10 225.00  25.478603 0.8338733 26.77533 -61031.25 -64605.89 26.17117 21.60424 19.97039
11 250.00  39.736493 0.7990719 25.77115 -61690.35 -64095.00 27.15694 22.54515 20.91232
12 275.00  59.431251 0.7592362 24.86701 -62370.65 -63557.52 28.14000 23.72806 22.35126
13 300.00  85.837843 0.7124075 24.04945 -63071.13 -62980.94 29.14072 25.28777 24.73943
14 325.00 120.457572 0.6545772 23.30725 -63790.84 -62341.39 30.19520 27.52189 29.44748
15 350.00 165.211289 0.5746875 22.63103 -64528.89 -61575.58 31.39713 31.34782 43.59852
```

The columns in the output are temperature (°C), pressure (bar), density of water (g cm^{-3}), logarithm of the equilibrium constant (only meaningful for reactions; see below), and standard molal Gibbs energy and enthalpy of formation from the elements (cal mol^{-1}), and standard molal entropy (cal K^{-1} mol^{-1}), volume (cm^3 mol^{-1}) and heat capacity (cal K^{-1} mol^{-1}).

Compared to other species available in CHNOSZ, the equations for calculating the properties of liquid H_2O are quite complex. The package uses a Fortran subroutine taken from SUPCRT for these calculations. See `help(water)` for more information.

4.2 A reaction

To calculate the properties of a reaction, enter the stoichiometric reaction coefficients as a second argument to `subcrt()`. Reactants have negative coefficients, and products have positive coefficients. The function call below also shows the specification of temperature.

```
> subcrt(c("C2H5OH","O2","CO2","H2O"),c(-1,-3,2,3),T=37)

info.character: found C2H5OH(aq), also available in liq, gas
info.character: found O2(aq), also available in gas
info.character: found CO2(aq), also available in gas
info.character: found H2O(liq), also available in gas
subcrt: 4 species at 310.15 K and 1 bar (wet)
$reaction
    coeff    name formula state ispecies
112    -1 ethanol  C2H5OH    aq      112
67     -3      O2      O2    aq       67
69      2     CO2     CO2    aq       69
1       3   water     H2O   liq        1

$out
      logK       G         H         S         V        Cp
1 227.5908 -322986 -326236.2 -10.43305 -26.46463 -66.17582
```

For historical reasons (i.e., the prevalence of the use of oxygen fugacity in geochemical modeling; Anderson, 2005), O_2 breaks the general rule in CHNOSZ that species whose states are not specified are given the aqueous designation if it is available in the thermodynamic database. If you want to specify the physical states of the species in the reaction, that's possible too. For example, we can ensure that dissolved O_2 instead of the gaseous form is used in the calculation.

```
> subcrt(c("C2H5OH","O2","CO2","H2O"),c(-1,-3,2,3),c("aq","aq","aq","liq"),T=37)

subcrt: 4 species at 310.15 K and 1 bar (wet)
$reaction
    coeff    name formula state ispecies
112    -1 ethanol  C2H5OH    aq      112
67     -3      O2      O2    aq       67
69      2     CO2     CO2    aq       69
1       3   water     H2O   liq        1

$out
      logK       G         H         S         V        Cp
1 227.5908 -322986 -326236.2 -10.43305 -26.46463 -66.17582
```

A useful feature of `subcrt()` is that it emits a warning if the reaction is not balanced. Let's say you forgot to account for oxygen on the left-hand side of the reaction[2].

```
> subcrt(c("C2H5OH","CO2","H2O"),c(-1,2,3),T=37)

info.character: found C2H5OH(aq), also available in liq, gas
info.character: found CO2(aq), also available in gas
info.character: found H2O(liq), also available in gas
subcrt: 3 species at 310.15 K and 1 bar (wet)
```

[2]This example is motivated by the unbalanced reaction found at the Wikipedia entry on ethanol metabolism on 2010-09-23 and still present as of 2011-08-15: "Complete Reaction: C_2H_6O(Ethanol)→C_2H_4O(Acetaldehyde)→$C_2H_4O_2$(acetic Acid) →Acetyl-CoA→$3H_2O$+$2CO_2$".

```
subcrt: reaction is not balanced; it is missing this composition:
 O
-6
$reaction
    coeff    name formula state ispecies
112    -1 ethanol  C2H5OH    aq      112
69      2     CO2     CO2    aq       69
1       3   water     H2O   liq        1

$out
      logK         G       H        S        V       Cp
1 219.9202 -312100.3 -333009 74.02581 67.43932 88.28986
```

The function still reports the results of the calculations, but use them very cautiously (only if you have a specific reason for writing an unbalanced reaction). In the next section we'll see how to use another feature of CHNOSZ to automatically balance reactions.

5 Basis species

5.1 What are basis species?

Basis species are a minimal number of chemical species that represent the compositional variation in a system. Operationally, a **system** is the combination of basis species and species of interest which is set up by the user to investigate a real-life system. The basis species are akin to thermodynamic components, but can include charged species.

There are at least two reasons to define the basis species when using CHNOSZ. First, you might want to use them to automatically balance reactions. Second, they are required for making chemical activity diagrams. Let's start with an example that *doesn't* work.

```
> basis(c("CO2","H2O","NH3","H2S","H+"))

    Error in put.basis(basis, mystates) :
      the stoichiometric matrix must be square and invertible
    In addition: Warning messages:
    1: basis: 5 compounds ( CO2 H2O NH3 H2S H+ )
    2: basis: 6 elements ( C H N O S Z )
```

A limitation of CHNOSZ is that the number of basis species must be equal to the number of elements, plus one if charge is present. This way, any possible species of interest made up of these elements can be compositionally represented by a linear combination of the basis species. Now let's write a working basis definition.

```
> basis(c("CO2","H2O","NH3","O2","H2S","H+"))

    C H N O S Z ispecies logact state
CO2 1 0 0 2 0 0       69      0    aq
H2O 0 2 0 1 0 0        1      0   liq
NH3 0 3 1 0 0 0       68      0    aq
O2  0 0 0 2 0 0       67      0    aq
H2S 0 2 0 0 1 0       70      0    aq
H+  0 1 0 0 0 1        3      0    aq
```

First basis definition! Note the column names, which give CHNOSZ its name. These represent the elements in the commonly-occurring amino acids, together with charge, denoted by "Z".

5.2 Auto-balancing a reaction

Now that the basis species are defined, try the unbalanced reaction again.

```
> subcrt(c("C2H5OH","CO2","H2O"),c(-1,2,3),T=37)

info.character: found C2H5OH(aq), also available in liq, gas
info.character: found CO2(aq), also available in gas
info.character: found H2O(liq), also available in gas
subcrt: 3 species at 310.15 K and 1 bar (wet)
subcrt: reaction is not balanced; it is missing this composition:
 O
-6
subcrt: adding missing composition from basis definition and restarting...
subcrt: 4 species at 310.15 K and 1 bar (wet)
$reaction
    coeff    name formula state ispecies
112    -1 ethanol  C2H5OH    aq      112
69      2     CO2     CO2    aq       69
1       3   water     H2O   liq        1
67     -3      O2      O2    aq       67

$out
      logK       G         H         S         V        Cp
1 227.5908 -322986 -326236.2 -10.43305 -26.46463 -66.17582
```

Here, `subcrt()` detected an unbalanced reaction, but since the missing element was among the elements of the basis species, it added the appropriate amount of $O_{2(gas)}$ to the reaction before running the calculations. You can go even further and eliminate CO_2 and H_2O from the function call, but still get the same results.

```
> subcrt(c("C2H5OH"),c(-1),T=37)

info.character: found C2H5OH(aq), also available in liq, gas
subcrt: 1 species at 310.15 K and 1 bar (wet)
subcrt: reaction is not balanced; it is missing this composition:
C H O
2 6 1
subcrt: adding missing composition from basis definition and restarting...
subcrt: 4 species at 310.15 K and 1 bar (wet)
$reaction
    coeff    name formula state ispecies
112    -1 ethanol  C2H5OH    aq      112
69      2     CO2     CO2    aq       69
1       3   water     H2O   liq        1
67     -3      O2      O2    aq       67

$out
      logK       G         H         S         V        Cp
1 227.5908 -322986 -326236.2 -10.43305 -26.46463 -66.17582
```

What if you were interested in the thermodynamic properties of the reaction of ethanol to acetaldehyde, but didn't want to balance the reaction yourself (and you also didn't know how the formulas of the species are written in the database)?

```
> subcrt(c("ethanol","acetaldehyde"),c(-1,1),T=37)
```

```
info.character: found ethanol(aq), also available in liq, gas
subcrt: 2 species at 310.15 K and 1 bar (wet)
subcrt: reaction is not balanced; it is missing this composition:
H
2
subcrt: adding missing composition from basis definition and restarting...
subcrt: 4 species at 310.15 K and 1 bar (wet)
$reaction
    coeff         name formula state ispecies
112  -1.0      ethanol  C2H5OH    aq      112
256   1.0 acetaldehyde  CH3CHO    aq      256
1     1.0        water     H2O   liq        1
67   -0.5           O2      O2    aq       67

$out
      logK         G         H          S         V        Cp
1 34.39068 -48805.62 -49023.41 -0.6110824 -7.942422 -34.07962
```

Notice how 2 H's needed to be added to the right-hand side of the reaction; in our definition of basis species this comes out to $H_2O - 0.5O_2$. With a different choice of basis species, but the same elements, the reaction might look quite different. As an example, suppose you had amino acids in mind. The first line below, `data(thermo)`, is a quick way to reset the `thermo` object to its original state, in order to clear the current system definition.

```
> data(thermo)
> basis(c("glutamic acid","methionine","isoleucine","lysine","tyrosine","H+"))

          C  H N O S Z ispecies logact state
C5H9NO4   5  9 1 4 0 0     1514      0    aq
C5H11NO2S 5 11 1 2 1 0     1525      0    aq
C6H13NO2  6 13 1 2 0 0     1520      0    aq
C6H14N2O2 6 14 2 2 0 0     1522      0    aq
C9H11NO3  9 11 1 3 0 0     1531      0    aq
H+        0  1 0 0 0 1        3      0    aq

> subcrt(c("ethanol","acetaldehyde"),c(-1,1),T=37)

info.character: found ethanol(aq), also available in liq, gas
subcrt: 2 species at 310.15 K and 1 bar (wet)
subcrt: reaction is not balanced; it is missing this composition:
H
2
subcrt: adding missing composition from basis definition and restarting...
subcrt: 5 species at 310.15 K and 1 bar (wet)
$reaction
      coeff         name   formula state ispecies
112  -1.000      ethanol    C2H5OH    aq      112
256   1.000 acetaldehyde    CH3CHO    aq      256
1520  0.500   isoleucine  C6H13NO2    aq     1520
1522 -0.125       lysine C6H14N2O2    aq     1522
1531 -0.250     tyrosine  C9H11NO3    aq     1531

$out
       logK        G        H          S         V        Cp
1 -1.341659 1904.018 1703.277 -0.5291135 -2.397983 -10.05446
```

In this case, the function finds that 2 H's are the compositional equivalent of $0.5C_6H_{13}NO_2 - 0.125C_6H_{14}N_2O_2 - 0.250C_9H_{11}NO_3$. It's pretty easy for the computer to figure that out using matrix operations, but probably isn't something you'd want to do by hand. You might complain that this reaction is not likely to represent an actual metabolic process ... as always, the challenge (and fun) of coming up with a useful basis definition is in relating the species to observable quantities.

5.3 It works for proteins too!

Let's set the basis definition again, this time using a keyword that refers to a preset combination of basis species commonly encountered in the documentation for CHNOSZ. Then we will use `subcrt()` to calculate the thermodynamic properties of a reaction to form a protein from the basis species.

```
> data(thermo)
> basis("CHNOS+")

    C H N O S Z ispecies logact state
CO2 1 0 0 2 0 0       69     -3    aq
H2O 0 2 0 1 0 0        1      0   liq
NH3 0 3 1 0 0 0       68     -4    aq
H2S 0 2 0 0 1 0       70     -7    aq
O2  0 0 0 2 0 0     3095    -80   gas
H+  0 1 0 0 0 1        3     -7    aq

> subcrt("LYSC_CHICK",1,T=25)

aa2eos: found LYSC_CHICK (C613H959N193O185S10, 129 residues)
subcrt: 1 species at 298.15 K and 1 bar (wet)
subcrt: reaction is not balanced; it is missing this composition:
   C    H    N    O    S
-613 -959 -193 -185  -10
subcrt: adding missing composition from basis definition and restarting...
subcrt: 6 species at 298.15 K and 1 bar (wet)
$reaction
      coeff       name             formula state ispecies
3369    1.0 LYSC_CHICK C613H959N193O185S10    aq     3369
69   -613.0        CO2                CO2    aq       69
1    -180.0      water                H2O   liq        1
68   -193.0        NH3                NH3    aq       68
70    -10.0        H2S                H2S    aq       70
3095  610.5     oxygen                 O2   gas     3095

$out
       logK        G        H        S         V        Cp
1 -46862.54 63931949 66481563 8601.824 -18320.13 -27314.51
```

Note that using the keyword argument in `basis()` also set the logarithms of activities (or fugacity in the case of $O_{2(g)}$) to nominal values. While these settings do not affect the results of the `subcrt()` calculation (which normally returns only the standard molal properties of the reaction), they are essential for calculating the relative stabilities of the species of interest.

If the protein is not available in CHNOSZ's own database, the amino acid composition of the protein can be retrieved from the UniProtKB (if the computer is connected to the internet). N.B. The following example is not evaluated when compiling this vignette in case the R package checks are run without internet access.

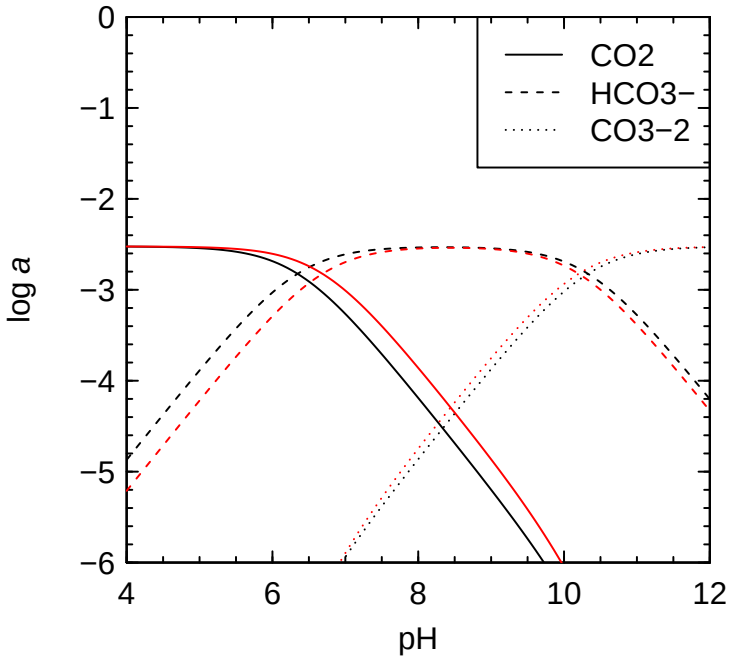
```
> aa <- uniprot.aa("ALAT1_HUMAN")
> add.protein(aa)
> subcrt("ALAT1_HUMAN",1,T=25)
```

6 Activity diagrams

6.1 Carbonate speciation (Bjerrum diagram)

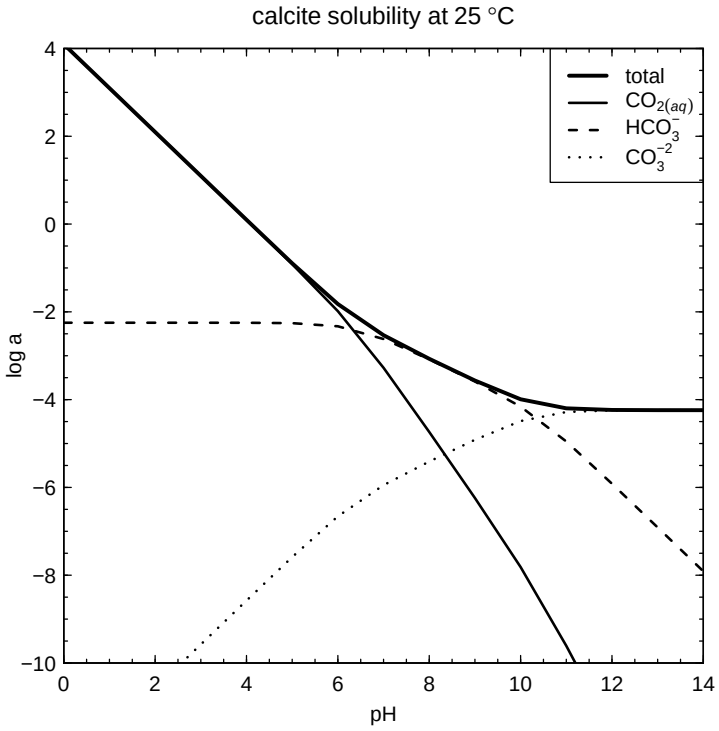
The sequence of commands `basis-species-affinity-[equilibrate]-diagram`, with various arguments, can be used to create a wide variety of diagrams. The first two lines below configure the basis species and the aqueous species. Some of the elements in the basis species are not in the species of interest but that's OK (the opposite wouldn't be). We want to make an activity diagram as a function of a single variable, and configure the program to do this by the single `pH` argument to `affinity()`. The `ylim` in the first `diagram()` is optional; it "zooms in" the y-axis (by default, the y-axis is expanded to contain all the values on the lines). Also the last two lines are optional, unless you really do want to see the effect of temperature (I do!)

```
> basis("CHNOS+")

    C H N O S Z ispecies logact state
CO2 1 0 0 2 0 0       69     -3    aq
H2O 0 2 0 1 0 0        1      0   liq
NH3 0 3 1 0 0 0       68     -4    aq
H2S 0 2 0 0 1 0       70     -7    aq
O2  0 0 0 2 0 0     3095    -80   gas
H+  0 1 0 0 0 1        3     -7    aq

> species(c("CO2", "HCO3-", "CO3-2"))

  CO2 H2O NH3 H2S O2 H+ ispecies logact state  name
1   1   0   0   0  0  0       69     -3    aq   CO2
2   1   1   0   0  0 -1       13     -3    aq HCO3-
3   1   1   0   0  0 -2       14     -3    aq CO3-2

> a <- affinity(pH=c(4, 12))

energy.args: temperature is 25 C
energy.args: pressure is Psat
energy.args: variable 1 is pH at 128 values from 4 to 12
subcrt: 9 species at 298.15 K and 1 bar (wet)

> e <- equilibrate(a)

balance: coefficients are moles of CO2 in formation reactions
equilibrate: balancing coefficients are 1 1 1
equilibrate: logarithm of total moles of CO2 is -2.52287874528034

> diagram(e, ylim=c(-6, 0))
> a <- affinity(pH=c(4, 12), T=150)

energy.args: temperature is 150 C
energy.args: pressure is Psat
energy.args: variable 1 is pH at 128 values from 4 to 12
subcrt: 9 species at 423.15 K and 4.76 bar (wet)

> e <- equilibrate(a)

balance: coefficients are moles of CO2 in formation reactions
equilibrate: balancing coefficients are 1 1 1
equilibrate: logarithm of total moles of CO2 is -2.52287874528034

> diagram(e, add=TRUE, col="red")
```



This just shows the speciation (relative abundances) of the aqueous carbonate species. Calculating the solubility of a carbonate mineral, or of CO_2 gas, is possible, but more involved; try the following:

```
> demo("solubility", ask=FALSE)
```



See the code of the demo (look for `demo/solubility.R` in the directory where CHNOSZ is installed) to change the calculation from calcite to CO_2.

6.2 Stability diagram for proteins

Suppose that we are asked to calculate the relative stabilities of some proteins from different organisms. We will use part of a case study from Dick (2008). *Methanocaldococcus jannaschii* is a hyperthermophilic methanogen known to live at higher temperatures than *Methanococcus voltae* (also a methanogen) and *Haloarcula japonica* (a halophile). These archaeal organisms produce cell-surface glycoproteins (a.k.a. surface-layer proteins).

After defining the basis species we can define the **species of interest**, i.e. those proteins whose relative stabilities we wish to calculate.

```
> basis("CHNOS")
```

```
    C H N O S ispecies logact state
CO2 1 0 0 2 0       69     -3    aq
H2O 0 2 0 1 0        1      0   liq
NH3 0 3 1 0 0       68     -4    aq
H2S 0 2 0 0 1       70     -7    aq
O2  0 0 0 2 0     3095    -80   gas
```

```
> species(c("SLAP_ACEKI", "CSG_METJA", "CSG_METVO", "CSG_HALJP"))
```

```
   CO2  H2O NH3 H2S      O2 ispecies logact state       name
1 3584 1431 926   4 -3730.5     3370     -3    aq SLAP_ACEKI
2 2555 1042 640  14 -2643.5     3371     -3    aq  CSG_METJA
3 2575 1070 645  11 -2668.0     3372     -3    aq  CSG_METVO
4 3669 1367 971   0 -3608.5     3373     -3    aq  CSG_HALJP
```

Note the output: the matrix denotes the coefficients of each of the basis species in the formation reaction for one mole of each of the species of interest. The **formation reaction** is the chemical reaction to form one mole of a species of interest (as a product) from a combination of basis species (as reactants and/or products, depending on the stoichiometric constraints). The formation reactions generally are *not* statements about the mechanisms of reactions. The species definition also includes reference values for the chemical activities of the species of interest.

Now we are all set up to calculate the chemical affinities of the formation reactions. The chemical affinity is the negative of the Gibbs energy change of a reaction per unit of reaction progress; it is calculated in CHNOSZ using $A = 2.303RT\log(K/Q)$ (R – gas constant, T – temperature, K – equilibrium constant, Q – activity product).

`affinity()` can accept arguments describing the range of chemical conditions we're interested in. The names of the arguments can refer to the basis species. Here, we vary the logarithm of the fugacity of oxygen. The chemical activities of the other basis species are taken to be constants equal to the values shown above.

```
> a <- affinity(O2=c(-90,-70))
```

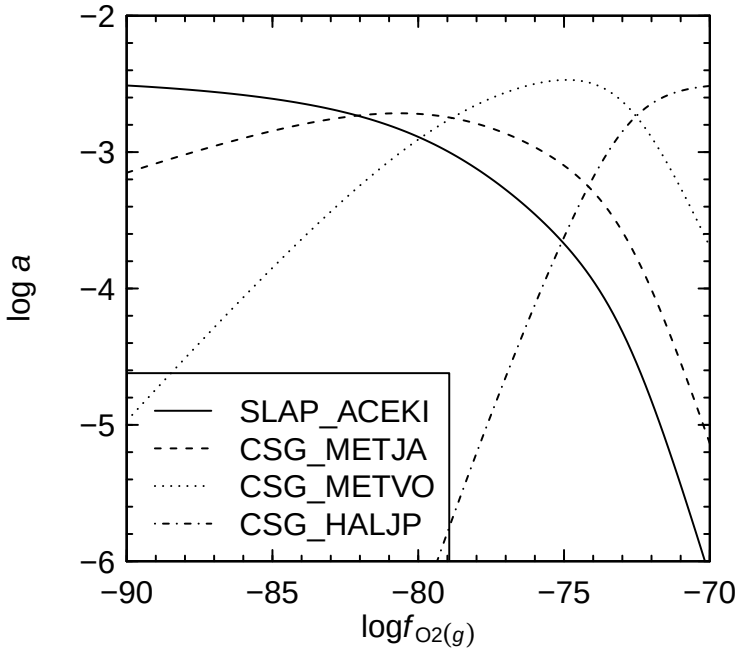
```
energy.args: temperature is 25 C
energy.args: pressure is Psat
energy.args: variable 1 is log_f(O2) at 128 values from -90 to -70
subcrt: 9 species at 298.15 K and 1 bar (wet)
```

Now we can use `equilibrate()` to calculate the equilibrium activities of the proteins and `diagram()` to plot them. `normalize=TRUE` invokes the normalization of the chemical formulas of the proteins by the lengths (number of residues), which in most cases is desirable for calculating equilibrium activities of proteins. We'll also specify where the legend should be placed on the plot.

```
> e <- equilibrate(a, normalize=TRUE)
```

```
balance: coefficients are protein length
equilibrate: balancing coefficients are 736 530 553 828
equilibrate: logarithm of total protein length is 0.422753941301348
equilibrate: using 'normalize' for molar formulas
```

```
> diagram(e, legend.x="bottomleft", ylim=c(-6, -2))
```



Notably, the protein from the organism found at the highest temperatures is relatively stable at more reduced conditions.

6.3 More proteins, more dimensions

Now let's add some bacterial surface-layer proteins. They are in some way functional analogs (but not homologs) of the archaeal cell-surface glycoproteins. This plot is made using the "maximum affinity method" to calculate the predominant species; alternatively we could insert a call to `equilibrate()` in order to calculate the equilibrium activities of all the species, but the resulting predominance diagram would be identical.

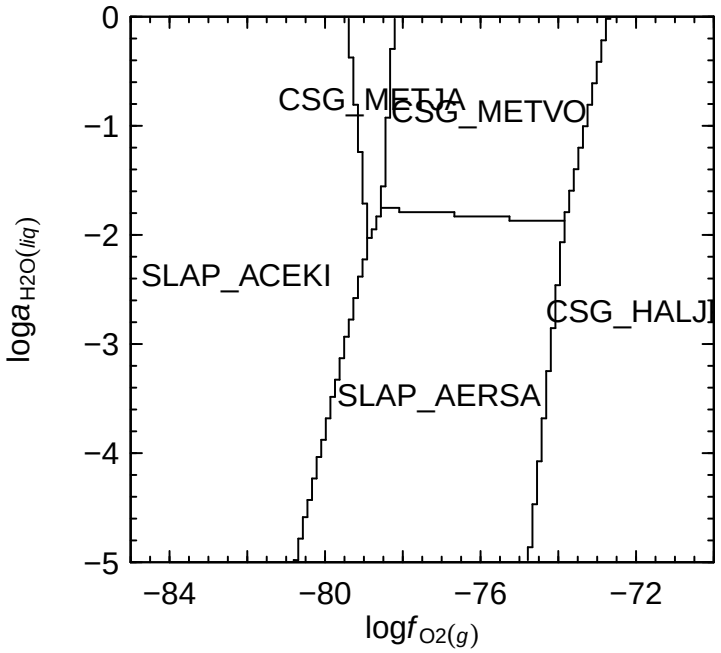
```
> species(c("SLAP_ACEKI", "SLAP_GEOSE", "SLAP_BACLI", "SLAP_AERSA"))
```

```
   CO2  H2O  NH3 H2S      O2 ispecies logact state       name
1 3584 1431  926   4 -3730.5     3370     -3    aq SLAP_ACEKI
2 2555 1042  640  14 -2643.5     3371     -3    aq  CSG_METJA
3 2575 1070  645  11 -2668.0     3372     -3    aq  CSG_METVO
4 3669 1367  971   0 -3608.5     3373     -3    aq  CSG_HALJP
5 5676 2320 1489   3 -5904.5     3374     -3    aq SLAP_GEOSE
6 3977 1594 1068   2 -4131.0     3375     -3    aq SLAP_BACLI
7 2250  861  618   2 -2322.5     3376     -3    aq SLAP_AERSA
```

```
> basis(c("NH3", "H2S"), c(-1, -10))
```

```
    C H N O S ispecies logact state
CO2 1 0 0 2 0       69     -3    aq
H2O 0 2 0 1 0        1      0   liq
NH3 0 3 1 0 0       68     -1    aq
H2S 0 2 0 0 1       70    -10    aq
O2  0 0 0 2 0     3095    -80   gas
```

```
> a <- affinity(O2=c(-85, -70), H2O=c(-5, 0))

energy.args: temperature is 25 C
energy.args: pressure is Psat
energy.args: variable 1 is log_f(O2) at 128 values from -85 to -70
energy.args: variable 2 is log_a(H2O) at 128 values from -5 to 0
subcrt: 12 species at 298.15 K and 1 bar (wet)

> diagram(a, normalize=TRUE)

balance: coefficients are protein length
diagram: plotting A/2.303RT from affinity(), divided by balancing coefficients
diagram: using 'normalize' in calculation of predominant species
```



Equilibrium predominances for proteins as a function of two chemical activities! If you don't like the colors in the plot, don't worry... the colors can be changed by using the `col` argument of `diagram()`. This example hints at the multidimensional nature of the stability problem. Note how the order of predominance fields at $\log a_{H_2O} = 0$ matches the order of proteins with highest equilibrium activities in the previous diagram. Interpreting the meaning of low activities of H_2O in these calculations remains a challenge.

Why did we increase the activity of NH_3 and decrease that of H_2S? It was done here in order to increase the size of the equilibrium predominance fields of the bacterial proteins. This behavior is a result of the elemental makeup of the proteins: the bacterial proteins under consideration are, per residue, more nitrogen-rich and sulfur-poor than their archaeal counterparts (except for CSG_HALJP, which has no sulfur). CHNOSZ has a function to display the compositional makeup of the proteins, per residue (indicated by `normalize=TRUE`), in terms of the basis species.

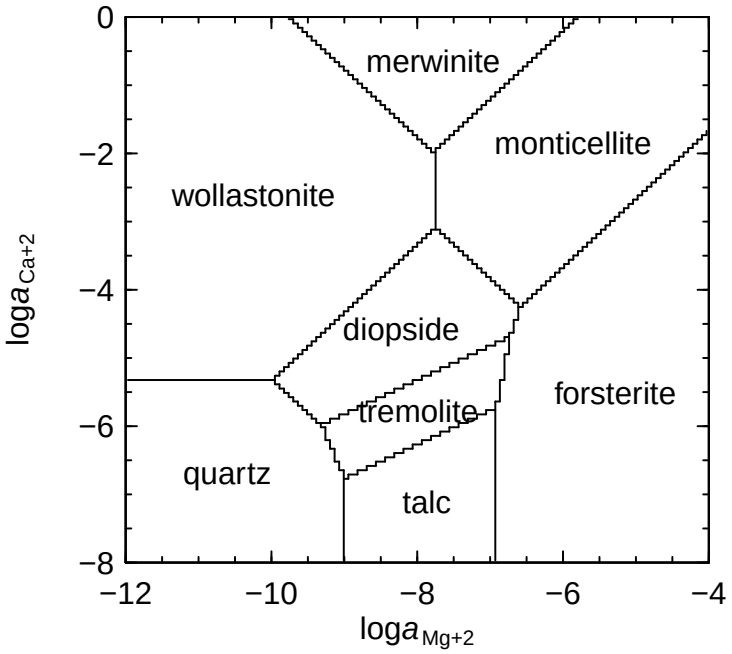
```
> protein.basis(species()$name, normalize=TRUE)

          CO2      H2O      NH3         H2S        O2
[1,] 4.869565 1.944293 1.258152 0.005434783 -5.068614
[2,] 4.820755 1.966038 1.207547 0.026415094 -4.987736
[3,] 4.656420 1.934901 1.166365 0.019891501 -4.824593
```

```
[4,] 4.431159 1.650966 1.172705 0.000000000 -4.358092
[5,] 4.737896 1.936561 1.242905 0.002504174 -4.928631
[6,] 4.712085 1.888626 1.265403 0.002369668 -4.894550
[7,] 4.677755 1.790021 1.284823 0.004158004 -4.828482
```

6.4 A mineral example

This example is modeled after a figure on p. 246 of Bowers et al. (1984) for the system HCl-H2O-CaO-CO2-MgO-(SiO2) at 300 °C and 1000 bar.

```
> basis(c("HCl","H2O","Ca+2","CO2","Mg+2","SiO2","O2","H+"),
+   c(999,0,999,999,999,999,999,-7))

     C Ca Cl H Mg O Si Z ispecies logact state
HCl  0  0  1 1  0 0  0 0      883    999    aq
H2O  0  0  0 2  0 1  0 0        1      0   liq
Ca+2 0  1  0 0  0 0  0 2       10    999    aq
CO2  1  0  0 0  0 2  0 0       69    999    aq
Mg+2 0  0  0 0  1 0  0 2        9    999    aq
SiO2 0  0  0 0  0 2  1 0       72    999    aq
O2   0  0  0 0  0 2  0 0       67    999    aq
H+   0  0  0 1  0 0  0 1        3     -7    aq

> species(c("quartz","talc","forsterite","tremolite","diopside",
+   "wollastonite","monticellite","merwinite"))

  HCl H2O Ca+2 CO2 Mg+2 SiO2 O2  H+ ispecies logact state         name
1   0   0    0   0    0    1  0   0     2014      0   cr1       quartz
2   0   4    0   0    3    4  0  -6     2039      0    cr         talc
3   0   2    0   0    2    1  0  -4     1929      0    cr   forsterite
4   0   8    2   0    5    8  0 -14     2041      0    cr    tremolite
5   0   2    1   0    1    2  0  -4     1900      0    cr     diopside
6   0   1    1   0    0    1  0  -2     2043      0    cr wollastonite
7   0   2    1   0    1    1  0  -4     1985      0    cr monticellite
8   0   4    3   0    1    2  0  -8     1981      0    cr    merwinite

> a <- affinity("Mg+2"=c(-12,-4),"Ca+2"=c(-8,0),T=300,P=1000)

energy.args: temperature is 300 C
energy.args: pressure is 1000 bar
energy.args: variable 1 is log_a(Mg+2) at 128 values from -12 to -4
energy.args: variable 2 is log_a(Ca+2) at 128 values from -8 to 0
subcrt: 16 species at 573.15 K and 1000 bar (wet)

> diagram(a)

balance: coefficients are moles of SiO2 in formation reactions
diagram: plotting A/2.303RT from affinity(), divided by balancing coefficients
```



The 999's in the assignment of logarithms of activities of basis species could be any number – these settings do not affect the outcome of the calculation. This is so because 1) HCl, CO_2 and O_2 have zero stoichiometric coefficients in the species, 2) the activities of Ca^{+2} and Mg^{+2} correspond to the axes of the diagram, and their ranges are taken from the call to `affinity()`, and 3) SiO_2 is the immobile (conserved) component. Also note that "Mg+2" and "Ca+2" are not valid names of objects in R, but we can use them as names of arguments by putting them in quotation marks in the call to `affinity()`.

Here, the scales of the axes here depend on the pH setting. This calculation is therefore logically different from the formulation used by Bowers et al. (1984), where the axes are $\log\left(a_{Mg^{+2}}/\sigma_{Mg^{+2}}a_{H^+}^2\right)$ and $\log\left(a_{Ca^{+2}}/\sigma_{Ca^{+2}}a_{H^+}^2\right)$ (where σ is a function of the solvation of the subscripted species). However, the geometry of the stability fields in the diagram produced here is consistent with the previous work.

In just a few lines it's possible to make a wide variety of activity diagrams for organic and inorganic species. Try it for your favorite system!

7 Where to go from here

You can explore the package documentation through R's help system; just type `help.start()` at the command line and select CHNOSZ in the browser window that comes up. Besides this document, there are other vignettes on topics of relative abundances and chemical activities of proteins, relative protein stability in a hot spring, and Gibbs energy minimization.

As a more visual way to get an idea of the types of calculations available in CHNOSZ, try running the examples in the help files for individual functions. A good one to try out might be `diagram()`; you can run all of the examples there with a single command:

```
> example(diagram)
```

Or you can use the following to run *all* of the examples provided in the documentation for the package. You will see a lot of text fly by on the screen, as well as a variety of plots. The examples will take about 5–10 minutes to run, depending on your machine.

```
> examples()
```

There are even more examples that can be accessed by `demo()` (or `demos()` to run all of them):

```
> demo("findit")
```

If you want to add to or modify the thermodynamic database, read the instructions at the top of the help page for `thermo`:

```
> help(thermo)
```

Have fun!

8 More activity diagrams

The following pages contain activity diagrams created with more complex series of command to show what "real life" usage of CHNOSZ might look like. Also, examples using some functions not covered above (buffer, revisit, findit) are included. To save space, the output from the commands (other than the plots) is hidden.

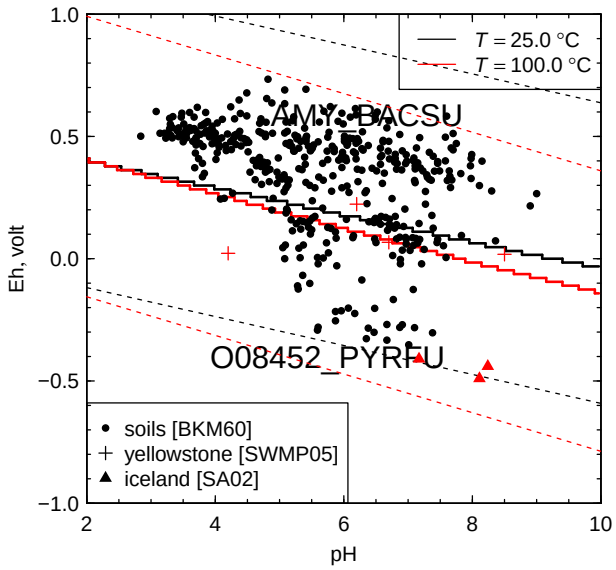
8.1 Protein Eh-pH

This plot, showing the relative stabilities of extracellular α-amylase on an Eh-pH diagram is similar to Fig. 13 of Dick et al. (2006). The basis("CHNOSe") includes an electron, so Eh can be used as a variable. The arguments in `diagram()` include setting the balanced component to CO_2; `normalize=FALSE` is left as the default so that whole protein formulas, not normalized by protein length, are used in the calculations.

Note that `add.obigt()` could be added at the very beginning to load the properties of the methionine sidechain group used by Dick et al. (2006) (the default values in the current version of CHNOSZ are taken from LaRowe and Dick, 2012), but for this example it doesn't make a whole lot of difference.

Toward the end of the script, points are added for Eh-pH values from soils (Baas Becking et al., 1960) and hot springs in Yellowstone (+) (Spear et al., 2005) and Iceland (▲) (Stefánsson and Arnórsson, 2002). The symbols identifying the latter two sources were swapped in the figure caption of Dick et al. (2006). Finally legends are drawn to identify the lines and symbols. The `describe.property()` function of CHNOSZ is used to generate the temperature notation (italic *T*, and degree sign in the units).

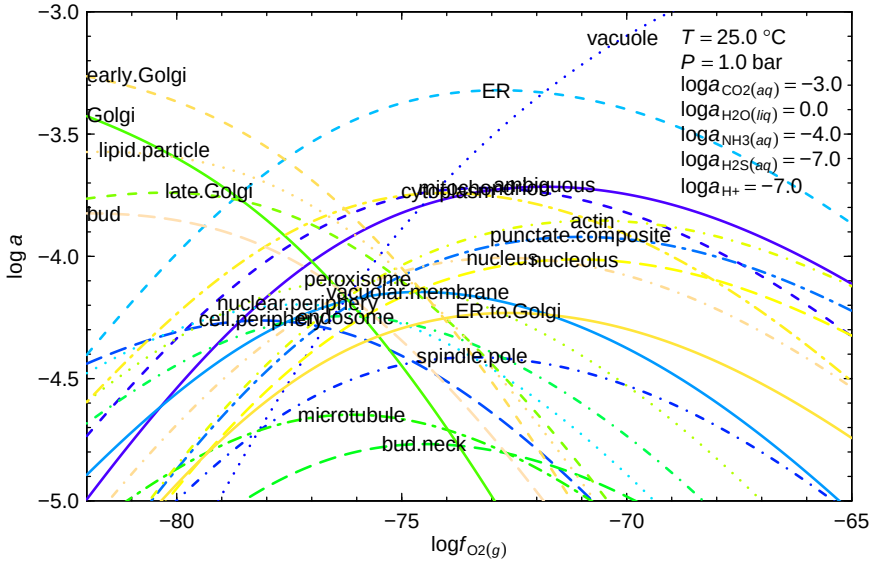
```
> basis("CHNOSe")
> basis(c("NH3", "H2S"), c(-6, -3))
> species(c("AMY_BACSU", "O08452_PYRFU"))
> a <- affinity(pH=c(2, 10), Eh=c(-1, 1))
> diagram(a, balance="CO2", lwd=2, fill=NULL, cex.names=1.5)
> a <- affinity(pH=c(2, 10), Eh=c(-1, 1), T=100)
> diagram(a, balance="CO2", lwd=2, fill=NULL, col="red", names=NULL, add=TRUE)
> water.lines()
> water.lines(T=398.15, col="red")
> BKMdat <- read.csv(system.file("extdata/cpetc/BKM60_Fig7.csv", package="CHNOSZ"))
> points(BKMdat$pH, BKMdat$Eh, pch=20)
> points(c(8.5, 6.2, 4.2, 6.7), c(0.018, 0.223, 0.022, 0.067), pch=3, col="red")
> points(c(8.24, 7.17, 8.11), c(-0.44, -0.41, -0.49), pch=17, col="red")
> ltext <- c(describe.property("T", 25), describe.property("T", 100))
> legend("topright", legend=ltext, lty=1, col=c("black", "red"))
> ltext <- c("soils [BKM60]", "yellowstone [SWMP05]", "iceland [SA02]")
> legend("bottomleft", legend=ltext, pch=c(20, 3, 17))
```



8.2 Subcellular proteins

Localizations and abundances of proteins from YeastGFP (Huh et al., 2003; Ghaemmaghami et al., 2003) are used here to calculate an abundance-weighted average of amino acid compositions of proteins in different subcellular compartments of yeast. The relative stabilities of these 23 model proteins are calculated as a function of the logarithm of oxygen fugacity ($\log f_{O_{2(g)}}$). This figure is similar to Fig. 3 of Dick (2009). Differences in positions of the lines can be attributed to updated parameters for the methionine sidechain group (LaRowe and Dick, 2012) that are used in the current version of CHNOSZ.

```
> locations <- yeastgfp()
> gfp <- yeastgfp(locations)
> aa <- more.aa(gfp$protein, "Sce")
> for(i in 1:length(locations)) {
+   avgaa <- aasum(aa[[i]], gfp$abundance[[i]], average=TRUE, protein=locations[i])
+   add.protein(avgaa)
+ }
> basis("CHNOS+")
> species(locations, "Sce")
> a <- affinity(O2=c(-82, -65))
> e <- equilibrate(a, loga.balance=0, normalize=TRUE)
> mycolor <- topo.colors(length(locations))
> diagram(e, names=locations, ylim=c(-5, -3), legend.x=NA,
+   col=mycolor, lwd=2)
> dp <- describe.property(c("T", "P"), c(25, 1))
> db <- describe.basis(ibasis=(1:6)[-5])
> legend("topright", legend=c(dp, db), bty="n")
```

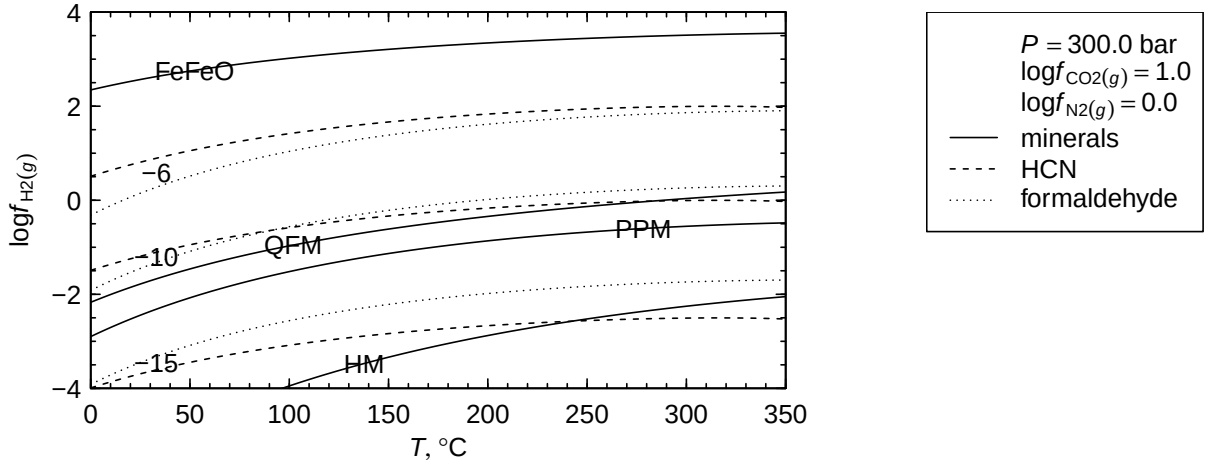


Notable features include: proteins in the early Golgi, endoplasmic reticulum (ER) and vacuole are chemically the most stable relative to those in other locations in the cell; proteins in the vacuole are very oxidized; the stability of proteins decreases going from early Golgi to Golgi to late Golgi; the least stable proteins (or most energetic) are found in the microtubules and bud neck.

8.3 Buffers

Chemical activity buffers permit calculating the activities of basis species from the activities of the species of interest, instead of the other way around. In CHNOSZ there are two ways to perform buffer calculations: by assigning the name of a buffer in `thermo$buffer` to the basis species, or by using the `what` argument of `diagram()` to solve for the activity of the indicated basis species. The former method is more versatile (multiple activities can be buffered, e.g. both S_2 and O_2 by pyrite-pyrrhotite-magnetite, and the buffers have effects on equilibrium activity diagrams) while the latter is more convenient (no need to set up the buffer in `thermo$buffer` – the buffers come from the species of interest). The plot below, based on Fig. 6 of Schulte and Shock (1995), shows values of $\log f_{H_{2(g)}}$ buffered by minerals and in equilibrium with different activities of organic species that are calculated using these two methods.

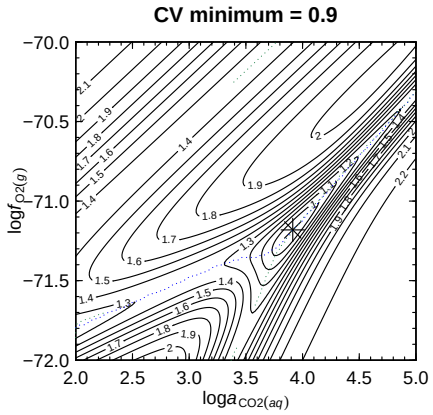
```
> layout(matrix(1:2, nrow=1), widths=c(2, 1))
> b.species <- c("Fe", "CO2", "H2O", "N2", "H2", "H2S", "SiO2")
> b.state <- c("cr1", "gas", "liq", "gas", "gas", "aq", "aq")
> b.logact <- c(0, 1, 0, 0, 0, 0, 0)
> basis(b.species, b.state, b.logact)
> xlim <- c(0, 350)
> thermo.plot.new(xlim=xlim, ylim=c(-4, 4), xlab=axis.label("T"), ylab=axis.label("H2"))
> bufferline <- function(buffer, ixlab) {
+   basis("H2", buffer)
+   a <- affinity(T=xlim, P=300, return.buffer=TRUE, exceed.Ttr=TRUE)
+   lines(a$vals[[1]], a$H2)
+   text(a$vals[[1]][ixlab], a$H2[ixlab], buffer)
+ }
> bufferline("FeFeO", 20)
> bufferline("QFM", 38)
> bufferline("PPM", 102)
> bufferline("HM", 51)
> basis("H2", 0)
> for(logact in c(-6, -10, -15)) {
+   species(c("formaldehyde", "HCN"), logact)
+   a <- affinity(T=xlim, P=300)
+   d <- diagram(a, what="H2", lty=c(2, 3), add=TRUE)
+   text(a$vals[[1]][13], mean(sapply(d$plotvals, c)[13, ]), logact)
+ }
> plot.new()
> legend("topleft", legend = c(describe.property("P", 300), describe.basis(ibasis=c(2,4)),
+   "minerals", "HCN", "formaldehyde"), lty=c(NA, NA, NA, 1, 2, 3), bg="white")
```



8.4 revisit

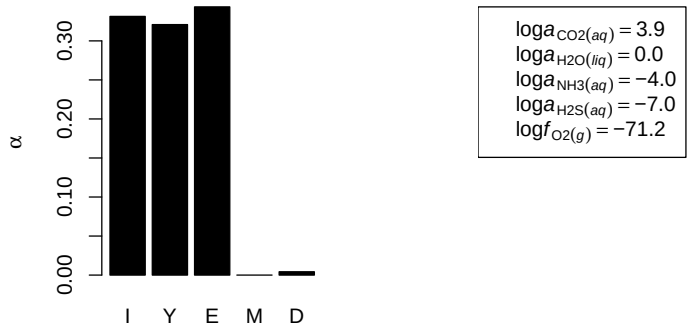
`revisit()` computes some summary statistics (default is the coefficient of variation) about the equilibrium chemical activities of species calculated by `equilibrate()`. In this example, the coefficient of variation of the activities of the amino acids is plotted as a function of $\log f_{O_{2(g)}}$ and $\log a_{CO_{2(aq)}}$.

```
> basis("CHNOS")
> species(c("isoleucine", "tyrosine", "glutamic acid", "methionine", "aspartic acid"))
> a <- affinity(CO2=c(2, 5), O2=c(-72, -70))
> e <- equilibrate(a, balance=1)
> r <- revisit(e)
> title(main=paste("CV minimum =", round(r$optimum, 2)))
```



The `balance=1` in `equilibrate()` means the relative stabilities are calculated using the formation reactions written per mole of amino acid (not conserving e.g. CO_2 which would be the default behaviour in this system). Note the star showing the conditions where the coefficient of variation is minimized. Let's look at the fractional equilibrium abundances of the amino acids at these conditions.

```
> basis(c("CO2", "O2"), c(r$x, r$y))
> a <- affinity()
> par(mfrow=c(1, 2))
> e <- equilibrate(a, balance=1)
> d <- diagram(e, alpha=TRUE, names=aminoacids(1, species()$name))
> plot.new()
> legend("topleft", describe.basis(basis()), bg="white")
```

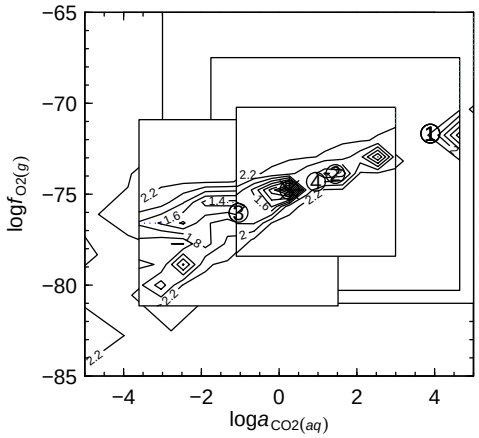


This hypothetical metastably equilibrated mixture has very little methionine and aspartic acid. Can we find where the relative abundances of the amino acids have a more even distribution?

8.5 findit

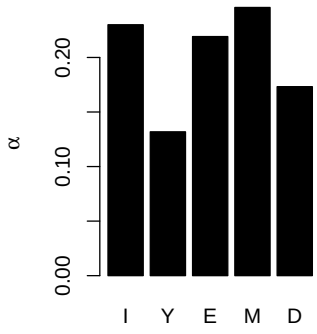
`findit()` performs a gridded search, on successively smaller hypercubes, for the conditions that optimize one of the statistics returned by `revisit()`. In this example containing five species, with `balance=1`, four compositional variables are the maximum that can be considered (NH_3 is excluded because it has a reaction coefficient of 1 in all of the formation reactions); the hypercube in this case is a tesseract. The grid resolution is 10, so the equilibrium chemical activities of the amino acids are computed at 10^4 discrete combinations of chemical potential of the basis species with each iteration.

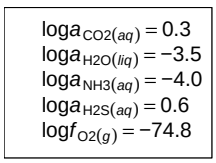
```
> basis("CHNOS")
> species(c("isoleucine", "tyrosine", "glutamic acid", "methionine", "aspartic acid"))
> f <- findit(list(CO2=c(-5, 5), O2=c(-85, -65), H2S=c(-10, 5), H2O=c(-10, 0)),
+   niter=5, res=10, balance=1)
```



After 5 iterations, what are the fractional equilibrium abundances of the amino acids? Note that, during its operation, `findit()` updates the activities of the basis species so we don't have to set them manually.

```
> a <- affinity()
> par(mfrow=c(1, 2))
> e <- equilibrate(a, balance=1)
> d <- diagram(e, alpha=TRUE, names=aminoacids(1, species()$name))
> plot.new()
> legend("topleft", describe.basis(basis()), bg="white")
```





We found a combination of chemical activities of basis species that lowered the variation of the equilibrium activities of the amino acids. Woohoo!

A more complete analysis might include more amino acids, different balancing constraints or chemical activity (or even temperature) limits, and would probably increase `niter` and/or `res` in `findit()`. Note that the results of a high-dimensional optimization such as this one may be misleading if the resolution is not high enough, or the initial hypercube is too small. Even after a successful simulation, it remains a matter of discussion what the optimized chemical activities of the basis species and the amino acids signify.

9 Document information

Revision history:

- 2010-09-30 Initial version
- 2011-08-15 Add `browse.refs()`; modifying database hint changed to `help(thermo)`
- 2012-06-16 Add "More activity diagrams"
- 2015-05-14 Add warning about internal consistency of thermodynamic data

R session information:

```
> sessionInfo()

R version 3.2.2 (2015-08-14)
Platform: x86_64-slackware-linux-gnu (64-bit)
Running under: Slackware 14.1

locale:
 [1] LC_CTYPE=en_US       LC_NUMERIC=C         LC_TIME=en_US        LC_COLLATE=C
 [5] LC_MONETARY=en_US    LC_MESSAGES=en_US    LC_PAPER=en_US       LC_NAME=C
 [9] LC_ADDRESS=C         LC_TELEPHONE=C       LC_MEASUREMENT=en_US LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] CHNOSZ_1.0.6 knitr_1.5

loaded via a namespace (and not attached):
[1] formatR_0.10     tools_3.2.2      codetools_0.2-14 highr_0.3        digest_0.6.8
[6] stringr_0.6.2    evaluate_0.5.1
```

References

G. M. Anderson. *Thermodynamics of Natural Systems*. Cambridge University Press, 2nd edition, 2005. URL http://www.worldcat.org/oclc/474880901.

L. G. M. Baas Becking, I. R. Kaplan, and D. Moore. Limits of the natural environment in terms of pH and oxidation-reduction potentials. *Journal of Geology*, 68(3):243–284, 1960. URL http://www.jstor.org/stable/30059218.

Teresa S. Bowers, Kenneth J. Jackson, and Harold C. Helgeson. *Equilibrium Activity Diagrams for Coexisting Minerals and Aqueous Solutions at Pressures and Temperatures to 5 kb and 600°C*. Springer-Verlag, Heidelberg, 1984. URL http://www.worldcat.org/oclc/11133620.

J. M. Dick, D. E. LaRowe, and H. C. Helgeson. Temperature, pressure, and electrochemical constraints on protein speciation: group additivity calculation of the standard molal thermodynamic properties of ionized unfolded proteins. *Biogeosciences*, 3(3):311–336, 2006. doi: 10.5194/bg-3-311-2006.

Jeffrey M. Dick. Calculation of the relative metastabilities of proteins using the CHNOSZ software package. *Geochemical Transactions*, 9:10, 2008. doi: 10.1186/1467-4866-9-10.

Jeffrey M. Dick. Calculation of the relative metastabilities of proteins in subcellular compartments of *Saccharomyces cerevisiae*. *BMC Systems Biology*, 3:75, 2009. doi: 10.1186/1752-0509-3-75.

Sina Ghaemmaghami, Won-Ki Huh, Kiowa Bower, Russell W. Howson, Archana Belle, Noah Dephoure, Erin K. O'Shea, and Jonathan S. Weissman. Global analysis of protein expression in yeast. *Nature*, 425 (6959):737 – 741, 2003. doi: 10.1038/nature02046.

Won-Ki Huh, James V. Falvo, Luke C. Gerke, Adam S. Carroll, Russell W. Howson, Jonathan S. Weissman, and Erin K. O'Shea. Global analysis of protein localization in budding yeast. *Nature*, 425(6959):686–691, 2003. doi: 10.1038/nature02026.

James W. Johnson, Eric H. Oelkers, and Harold C. Helgeson. SUPCRT92: A software package for calculating the standard molal thermodynamic properties of minerals, gases, aqueous species, and reactions from 1 to 5000 bar and 0 to 1000°C. *Computers & Geosciences*, 18(7):899 – 947, 1992. doi: 10.1016/0098-3004(92)90029-Q.

Douglas E. LaRowe and Jeffrey M. Dick. Calculation of the standard molal thermodynamic properties of crystalline peptides. *Geochimica et Cosmochimica Acta*, 80:70–91, 2012. doi: 10.1016/j.gca.2011.11.041.

Mitchell D. Schulte and Everett L. Shock. Thermodynamics of Strecker synthesis in hydrothermal systems. *Origins of Life and Evolution of the Biosphere*, 25(1-3):161 – 173, 1995. doi: 10.1007/BF01581580.

Everett L. Shock and Harold C. Helgeson. Calculation of the thermodynamic and transport properties of aqueous species at high pressures and temperatures: Standard partial molal properties of organic species. *Geochimica et Cosmochimica Acta*, 54(4):915 – 945, 1990. doi: 10.1016/0016-7037(90)90429-O.

John R. Spear, Jeffrey J. Walker, Thomas M. McCollom, and Norman R. Pace. Hydrogen and bioenergetics in the Yellowstone geothermal ecosystem. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2555 – 2560, 2005. doi: 10.1073.pnas.0409574102.

Andri Stefánsson and Stefán Arnórsson. Gas pressures and redox reactions in geothermal fluids in Iceland. *Chemical Geology*, 190:251 – 271, 2002. doi: 10.1016/S0009-2541(02)00119-5.