Winding journey down (in Gibbs energy)

Jeffrey M. Dick

July 4, 2013

1 Introduction

`wjd()` appeared in CHNOSZ version 0.9-9. It implements the steepest descent algorithm for Gibbs energy minimization described by White, Johnson and Dantzig, 1958. This vignette shows some examples of its usage. The help page for `wjd()` in the CHNOSZ documentation has details of the implementation that are not covered in this vignette. The functions demonstrated here are not intended for aqueous solutions or heterogeneous phase equilibria, which are covered in some reviews (e.g. Nordstrom et al., 1979).

The arguments of `wjd()` have default values to simulate the example problem given by White et al., 1958. In their words, "The example taken is the determination of the composition of the gases arising from the combustion of a stoichiometric mixture of hydrazine and oxygen at 3500 °K and a pressure of 750 psi (ln P_{atm} =3.932)." There are ten species included in this example. The value of temperature does not appear explicitly in the algorithm, or in the arguments to `wjd()`. Instead, the standard Gibbs energies of the species, at the given temperature, are provided in dimensionless form, i.e. $\Delta G^\circ / RT$. Note that White et al., 1958 use the terminology "free energy" and the notation F°. The term "Gibbs energy" and corresponding notation are used here.

2 Is it winding?

The "winding" in the title refers to the observation that the abundances of species during a Gibbs energy minimization often do not all change in a monotonic fashion. This is because at each iteration a new direction of steepest descent is calculated; this direction is a vector of changes of mole numbers of all species in the system and is subject to mass balance constraints as well as the steepest-descent criterion.

Let's run `wjd()` with its default settings and save the output.

```
> w <- wjd()
```

What are the most abundant species, and how many iterations were taken?

```
> # the order of species abundance
> oX <- order(w$X, decreasing=TRUE)
> # the stoichiometries of the two most abundant species
> w$A[head(oX,2),]

     H N O
[1,] 2 0 1
[2,] 0 2 0

> # the number of iterations
> niter <- length(w$lambda)
> niter

[1] 7
```

We find that H2O and N2 are the most abundant species, after 7 iterations. The fourth most-abundant species, HO, is discussed below.

Let's track their mole fractions through the iterations. Write a function that returns the mole fractions of a species, after a specified number of iterations.

```
> iterfun <- function(imax, i) {
    w <- wjd(imax=imax)
    x <- w$X[i]/sum(w$X)
    return(x)
 }
```

Then apply the function over the different numbers of iterations, from 0 (initial conditions) to 7, and plot the values.

```
> par(mfrow=c(1, 2))
> sa <- sapply(0:niter, iterfun, i=oX[1])
> plot(0:niter, sa, xlab="iteration", ylab=paste("x", f1))
> sa <- sapply(0:niter, iterfun, i=oX[4])
> plot(0:niter, sa, xlab="iteration", ylab=paste("x", f4))
```
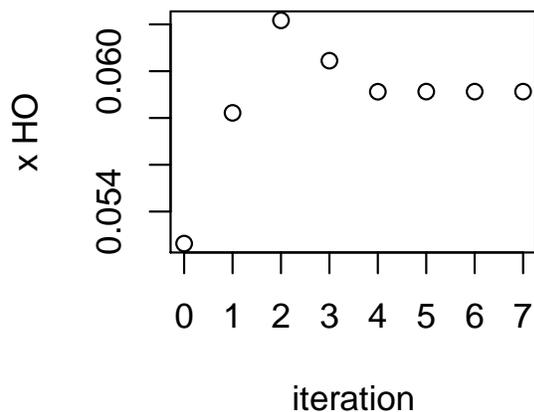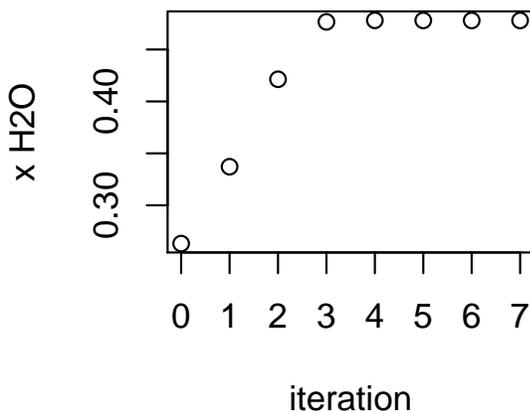

A bit of winding: the mole fraction of H2O generally increases (there is a very slight decrease at the fifth iteration), but the mole fraction of HO increases in the first two iterations, then decreases in following ones. This behavior is consistent with a decrease in Gibbs energy at *every* iteration; that can be verified by inspecting the values in the 7-iteration result:

```
> all(diff(w$F.Y) < 0)
```

```
[1] TRUE
```

The decrease in Gibbs energy becomes smaller with every iteration, as an equilibrium distribution of species is approached:

```
> identical(diff(w$F.Y), sort(diff(w$F.Y)))
```

```
[1] TRUE
```

2

3   Is it near the bottom? Testing for equilibrium

3.1   Change in Gibbs energy of the system becomes small

Equilibrium is synonymous with a global Gibbs energy minimum. Becoming convinced that the output from wjd() represents a near-equilibrium condition can be difficult. One type of observation that can be helpful is the amount of change during the iterations of the algorithm. As equilibrium is approached, it makes sense that the fractional decreases in Gibbs energy would become smaller and smaller. This is used as a stopping condition in the current implementation – if the fractional change, relative to the total energy of the system, reaches the value of Gfrac given in the arguments the iterations stop. Here are the fractional changes with each iteration:

```
> diff(w$F.Y)/w$F.Y[1:6]

[1] 1.929006e-02 3.555210e-03 5.841261e-04 9.726451e-06 2.465635e-06 1.155873e-07
[7] 2.317782e-10
```

Only the last value is below the default value of Gfrac in wjd().

3.2   Chemical potentials of the elements (from different species representing them) become more uniform

Equilibrium is also synonymous with uniformity of chemical potentials (μ) of the thermodynamic components throughout the system. For the default system in wjd(), and perhaps other systems of interest, the elements themselves are a conceivable set of the components. Often, the number and compositions of the species are such that multiple combinations of species satisfy the stoichiometric conditions necessary to be a basis set, and therefore can be used to compute the chemical potentials of the elements. At equilibrium, these different combinations of species would all yield the same chemical potentials of the elements.

There is a supporting function, element.potentials(), that computes the chemical potentials of the elements using different eligible combinations of species. It has an option to plot the results. Let's first look at a plot showing the results after 3 iterations.

```
> w3 <- wjd(imax=3)
> ep3 <- element.potentials(w3, plot.it=TRUE)
```

max difference (range) = 2.7

max difference (range) = 1.4

max difference (range) = 4
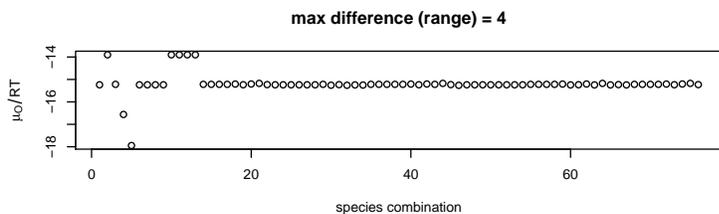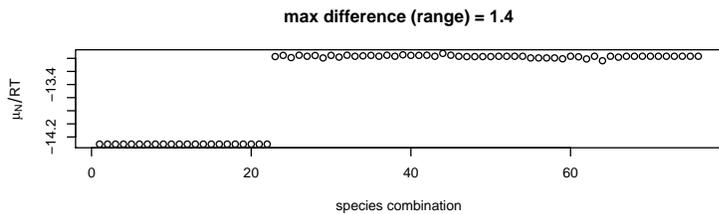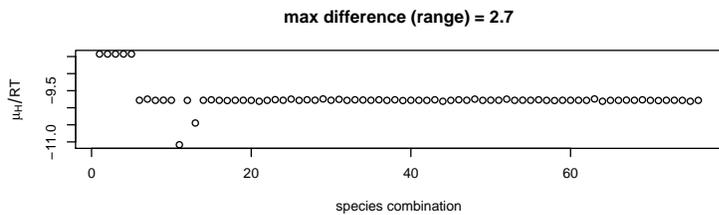
Here's the plot for the default settings of `wjd()` which takes 6 iterations:

```
> ep <- element.potentials(w, plot.it=TRUE)
```

max difference (range) = 1.5e−05

max difference (range) = 7.6e−06

max difference (range) = 2.3e−05
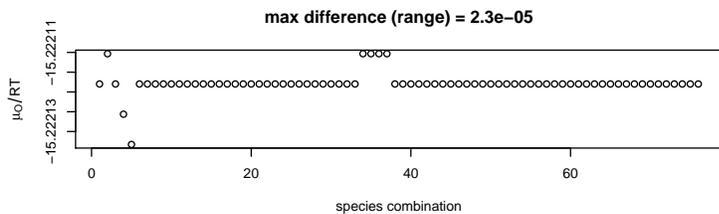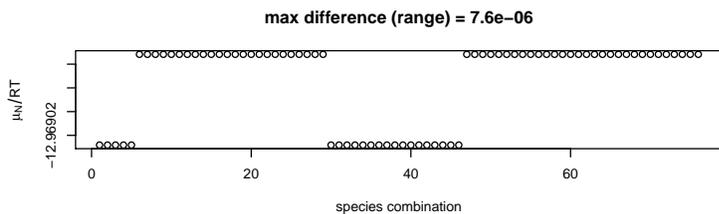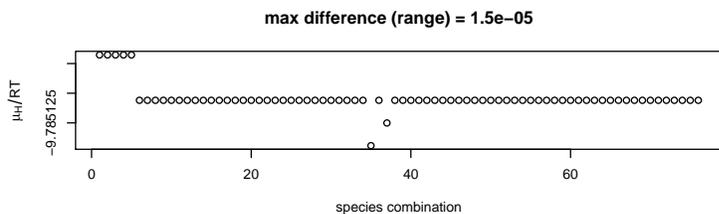
That one shows considerably less deviation than the first plot. The differences don't become zero, but perhaps they are small enough to accept as an operational solution. There is a function, `is.near.equil()`, that provides a logical type result based on the range of chemical potentials of the elements calculated from

4

the different eligible species combinations. It returns TRUE if the maximum difference for any element is below $\mu/RT = 0.01$, but this limit can be changed through the `tol` argument to the function. With the default setting, `is.near.equil()` does pass the 7-iteration result, but by reducing the tolerance, the tests can be made to fail:

```
> is.near.equil(w3) # 3 iterations

is.near.equil: solution has variation of 4.0414527550358 in mu/RT of O
[1] FALSE

> is.near.equil(w)  # 7 iterations

[1] TRUE

> is.near.equil(w, tol=0.00001)

is.near.equil: solution has variation of 2.29301401670767e-05 in mu/RT of O
[1] FALSE
```

3.3 Different initial solutions yield similar results

Different guesses for an underdetermined system Although it seems to be the case in the examples shown so far, increasing the number of iterations does not necessarily bring one closer to true equilibrium, since being trapped in a local energy minimum is a possibility. Sometimes local minima can be shown to exist by starting with different mole numbers of species, that still yield the same bulk composition (mole numbers of elements). `guess()` exists to facilitate this sort of investigation. Its input the stoichiometric matrix of species in the system (the default refers to the same system as in `wjd()`) and the bulk composition; it outputs one set or a series of sets of mole numbers of species that satisfy the bulk composition constraints. In general, this in an underdetermined problem, so there are usually an infinite number of possible solutions.

Using the default method ("stoich") a single guess or multiple guesses are returned based on stoichiometric constraints (see the help page for details as well as the comments in the function code itself). If the "central" method is chosen instead, the single guess returned is the central solution from linear optimization using the package limSolve. Let's compare the guesses made by `guess()` with the default (from White et al., 1958) initial composition in `wjd()` .

```
> as.list(args(wjd))$Y

c(0.1, 0.35, 0.5, 0.1, 0.35, 0.1, 0.1, 0.1, 0.1, 0.1)

> Y1 <- guess(method="stoich")
> Y1

 [1] 0.012 0.002 0.990 0.992 0.002 0.002 0.002 0.002 0.002 0.002

> #Y2 <- guess(method="central")
> #Y2
```

Now let's run the minimization using the different guesses, count the numbers of iterations, and test them for equilibrium.

```
> wY1 <- wjd(Y=Y1)
> niterY1 <- length(wY1$lambda)
> niterY1

[1] 9

> is.near.equil(wY1, tol=0.0001)
```

```
is.near.equil: solution has variation of 0.000641091319222298 in mu/RT of O
[1] FALSE
```

Using the first guess generated using the "stoich" method there were 2 more iterations than using the default initial solution in `wjd()`.

Do the different initial guesses actually give similar results?

```
> plot(1:10, w$X, xlab="species", ylab="mole fraction")
> points(1:10, wY1$X, pch=0)
> #points(1:10, wY2$X, pch=2)
```
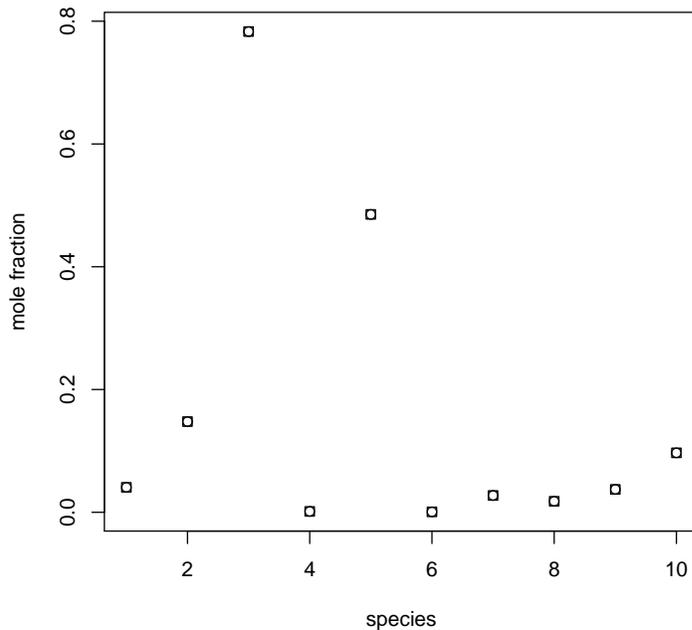

At that scale it's fairly difficult to tell them apart.

Comparing many different guesses   Let's test for closeness to equilibrium of all of the guesses available from `guess()` for the default set of species and bulk composition. First generate all of the guesses. Note that `guess()` returns NA for compositionally eligible combinations of species that have some negative mole numbers, so we filter those out.

```
> Ys <- guess(iguess=NULL, method="stoich")
> # total number of species combinations
> length(Ys)

[1] 76

> # species combinations that have all-positive mole numbers
> iYs <- which(!is.na(Ys))
> nguess <- length(iYs)
> nguess

[1] 39
```

Now let's run the minimization using each of those guesses, and test if each one `is.near.equil()`:

6

```
> sapply(iYs,function(i) is.near.equil(wjd(Y=Ys[[i]])))
```

```
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[17] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[33] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Looks like they are. What if we're really picky though and want to make sure the chemical potentials of the elements are very well-defined? We can decrease the tolerance of is.near.equil():

```
> sapply(iYs,function(i) is.near.equil(wjd(Y=Ys[[i]]),tol=0.0001))
```

```
is.near.equil: solution has variation of 0.000641091319222298 in mu/RT of O
is.near.equil: solution has variation of 0.00852890248348182 in mu/RT of O
is.near.equil: solution has variation of 0.000891875931088748 in mu/RT of O
is.near.equil: solution has variation of 0.000642445178012707 in mu/RT of O
is.near.equil: solution has variation of 0.000268593486817181 in mu/RT of O
is.near.equil: solution has variation of 0.00795288245710424 in mu/RT of O
is.near.equil: solution has variation of 0.000651868600300531 in mu/RT of O
is.near.equil: solution has variation of 0.00852890248347293 in mu/RT of O
is.near.equil: solution has variation of 0.000294556768547949 in mu/RT of O
is.near.equil: solution has variation of 0.00849629215749204 in mu/RT of O
is.near.equil: solution has variation of 0.00208418588385584 in mu/RT of O
is.near.equil: solution has variation of 0.00146234066616735 in mu/RT of O
is.near.equil: solution has variation of 0.000151054190643762 in mu/RT of O
is.near.equil: solution has variation of 0.000481312160735214 in mu/RT of O
is.near.equil: solution has variation of 0.000316491422417187 in mu/RT of O
is.near.equil: solution has variation of 0.00823996203233257 in mu/RT of O
is.near.equil: solution has variation of 0.004726265100046759 in mu/RT of O
 [1] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE
[14]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
[27]  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE
```

Well some of the tests failed. But maybe this is because we have stopped iterating too soon. Let's iterate until a we have smaller change in Gibbs energy of the system:

```
> sapply(iYs, function(i) {
    is.near.equil(wjd(Y=Ys[[i]], Gfrac=1e-9), tol=0.0001)
 })
```

```
is.near.equil: solution has variation of 0.000316491422417187 in mu/RT of O
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[14]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[27]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
```

We're back to passing all but one of the equilibrium tests based on uniformity of chemical potentials of the elements. What are the standard deviations of the resulting species abundances?

```
> Xs <- sapply(iYs, function(i) wjd(Y=Ys[[i]], Gfrac=1e-9)$X)
> apply(Xs, 1, sd)
```

```
 [1] 1.157303e-08 1.535826e-08 1.443102e-08 2.397691e-08 1.229067e-08 1.666586e-09
 [7] 8.164822e-09 2.567975e-09 8.879598e-09 5.216430e-09
```

Based on this exercise, starting from different initial species abundances for the same bulk composition, and arriving at a similar near-equilibrium solution, it seems likely that we are near the global minimum.

   (In fact, for ideal reactions in closed chemical systems there is only a unique solution (Othmer, 1976). Therefore, the algorithm should converge on the same result for any initial vector of species mole numbers with the same bulk composition.)

4 An example from the literature: prebiological atmospheres

We will try to reproduce a subset of the calculations of equilibrium between carbon-containing compounds in model prebiological atmospheres reported by Dayhoff et al. (1964), using standard Gibbs energies from Dayhoff et al. (1967). The standard Gibbs energies shown in Figure 2 of Dayhoff et al. (1967) for the selected compounds listed in Table 1 of Dayhoff et al. (1964) are provided with CHNOSZ. First let's have a look at what species there are:

```
> # read formulas and Gibbs energies
> file <- system.file("extdata/thermo/DLEN67.csv", package="CHNOSZ")
> dlen <- read.csv(file, stringsAsFactors=FALSE, row.names=1)
> t(dlen[, 1, drop=FALSE])
        water carbon dioxide nitrogen methane ammonia carbon monoxide ethane
formula "H2O" "CO2"          "N2"     "CH4"   "NH3"   "CO"            "C2H6"
        formic acid propane hydrogen cyanide methylamine butane  glycine    benzene
formula "CH2O2"     "C3H8"  "HCN"            "CH5N"      "C4H10" "C2H5NO2" "C6H6"
        hexane  naphthalene anthracene octane
formula "C6H14" "C10H8"     "C14H10"   "C8H18"
```

Now let's write some code to define the system. The bulk composition in C:H:O space has 40 percent oxygen, a percentage of carbon given by xC, and hydrogen as the remainder; the C:N ratio is taken to be 1:1 (Dayhoff et al., 1964). The temperature is 500 K and the pressure is 1 atmosphere.

```
> # turn formulas into a stoichiometric matrix
> A <- i2A(dlen$formula)
> # assemble Gibbs energies/RT at 500 K
> G0.RT <- 1000 * dlen$G500 / thermo$opt$R / thermo$opt$Tr
> # a function to minimize Gibbs energy for system with
> # given mole fraction of carbon (xC)
> min.atmos <- function(xC) {
  # the bulk composition C:H:N:O
  B <- c(xC, 100-40-xC, xC, 40)
  # guess the initial composition
  Y <- guess(A, B)
  w <- wjd(A=A, G0.RT=G0.RT, Y=Y, P=1, imax=90, Gfrac=1e-14)
  if(!is.near.equil(w)) cat(paste("not near equilibrium for xC=", xC, "\n"))
  return(w)
 }
```

Notice the increase in `imax` and the decrease in `Gfrac` from the defaults! These changes are needed to get closer to equilibrium (and we'll know we're not there if a message is shown). What does the prebiological atmosphere look like with 15% (not counting nitrogen) carbon?

```
> sort(min.atmos(15)$X, decreasing=TRUE)

is.near.equil: solution has variation of 8.92765537568458 in mu/RT of N
not near equilibrium for xC= 15
         H2O          CO2           N2          CH4          NH3           CO
1.624999e+01 1.187500e+01 7.499996e+00 3.124996e+00 8.375711e-06 1.712005e-06
        C2H6        CH2O2         C3H8         CH5N          HCN        C4H10
5.170619e-10 1.091612e-14 2.153007e-18 1.573453e-20 1.164057e-20 3.608739e-26
     C2H5NO2         C6H6        C6H14         C8H18        C10H8       C14H10
2.098217e-33 5.622884e-36 1.317368e-42 5.118320e-59 2.676105e-59 4.102533e-84
```

Well that's exciting! The order of abundances of H_2O, CO_2, CH_4, NH_3, CO, ethane (C_2H_6), formic acid (CH_2O_2) and so on matches the order on the left side of Figure 2 in Dayhoff et al. (1964). How about changing the carbon content?

```
> xCs <- seq(8, 47, 3)
> Xs <- sapply(xCs, function(xC) min.atmos(xC)$X)

is.near.equil: solution has variation of 0.0306270634092414 in mu/RT of N
not near equilibrium for xC= 8
is.near.equil: solution has variation of 7.27774296261396 in mu/RT of N
not near equilibrium for xC= 11
is.near.equil: solution has variation of 0.184746502791072 in mu/RT of C
not near equilibrium for xC= 14
is.near.equil: solution has variation of 1.02601836372492 in mu/RT of N
not near equilibrium for xC= 20
is.near.equil: solution has variation of 0.716037533741352 in mu/RT of C
not near equilibrium for xC= 23
is.near.equil: solution has variation of 4.08617998983382 in mu/RT of N
not near equilibrium for xC= 26
is.near.equil: solution has variation of 3.77616569840606 in mu/RT of N
not near equilibrium for xC= 29
is.near.equil: solution has variation of 0.181389575403315 in mu/RT of O
not near equilibrium for xC= 32
is.near.equil: solution has variation of 0.251980226156448 in mu/RT of O
not near equilibrium for xC= 38
is.near.equil: solution has variation of 0.389332461664541 in mu/RT of C
not near equilibrium for xC= 41
is.near.equil: solution has variation of 3.28731003660577 in mu/RT of O
not near equilibrium for xC= 44
is.near.equil: solution has variation of 16.396899839931 in mu/RT of O
not near equilibrium for xC= 47

> # normalize the mole numbers to mole fractions
> Xs <- t(t(Xs)/colSums(Xs))
> plot(-10, 0, xlim=c(0, 55), ylim=c(-25, 1), xlab="mole percent C", ylab="log10 mole fraction")
> for(i in 1:nrow(Xs)) lines(xCs, log10(Xs[i, ]))
> text(48, log10(Xs[, length(xCs)]), dlen$formula, adj=0)
> text(7, log10(Xs[, 1]), dlen$formula, adj=1)
```
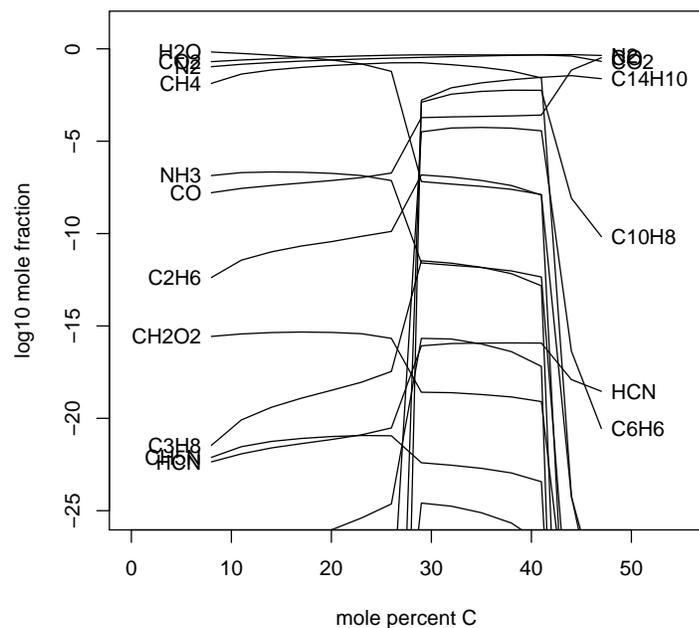
We triggered some "not near equilibrium" messages, but overall it seems well behaved (note that using the "central" method in guess() is one way that leads to fewer of these messages). And, as expected, it is similar to Fig. 2 of Dayhoff et al. (1964), with a major crossing of curves at about 28% carbon, together with an increase in aromatic compounds (e.g. naphthalene, anthracene) going toward higher carbon content. Unlike the figure in Dayhoff et al. (1964), there appears to be a second major crossing of curves at about 43% carbon, corresponding to a rise in CO. (For a higher resolution, try setting xCs <- seq(8, 47, 1).)

5 Running Down: Using a Thermodynamic Database

So far the examples shown here have been based on the chemical system defined by the default settings for the arguments to the functions. What if we're interested in a different system? It can be rather tedious to manually construct the stoichiometric matrices and vectors of standard Gibbs energies of the species. The function run.wjd() pulls the compositional and thermodynamic data for requested species from the thermodynamic database and then calls wjd().

5.1 At a single temperature

Let's start by finding the indices in the thermodynamic database of some liquid alkanes and aromatic compounds:

```
> alkanes <- c("n-hexane", "n-heptane", "n-octane", "n-nonane")
> ialk <- info(alkanes, "liq")
> aromatics <- c("naphthalene", "anthracene", "phenanthrene", "pyrene")
> iaro <- info(aromatics, "liq")
```

Let's find the equilibrium distribution of species for a bulk composition corresponding to a single mole of each species.

```
> waa <- run.wjd(c(ialk, iaro), Y=rep(1, 8))

subcrt: 8 species at 298.15 K and 1 bar
```

```
> waa$elements

  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
C   84   84   84   84   84   84   84   84   84    84
H  106  106  106  106  106  106  106  106  106   106

> is.near.equil(waa)

is.near.equil: solution has variation of 1.2356073555955 in mu/RT of H
[1] FALSE
```

The `waa$elements` shows that the bulk chemical composition in the Gibbs energy minimization was $C_{84}H_{106}$ (H/C = 1.261905). That FALSE means the chemical potentials of the elements calculated from different combinations of species differ by more than 0.01, the default setting of `tol` in `is.near.equil()`. We can make it TRUE by running more iterations and increasing the number of intervals tested for fractional distance change at each step (`nlambda`):

```
> waa <- run.wjd(c(ialk, iaro), Y=rep(1, 8), imax=20, Gfrac=1e-14, nlambda=501)

subcrt: 8 species at 298.15 K and 1 bar

> is.near.equil(waa)

[1] TRUE
```

Let's plot the abundances of the species:

```
> bp <- barplot(waa$X, ylab="moles")
> text(bp, rep(0.2,8), c(alkanes, aromatics), srt=90, adj=0)
```
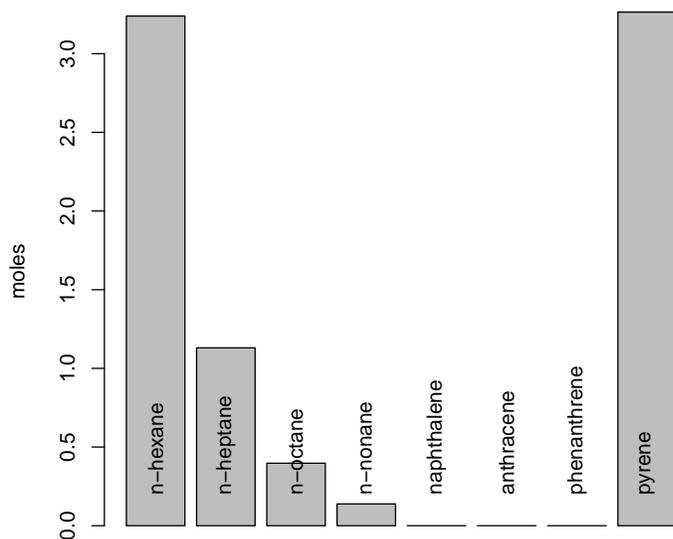

Now let's calculate the chemical potentials of the elements (dimensionless values, μ / RT) ...

```
> ep <- equil.potentials(waa)
> print(ep)
```

```
        C         H
5863.435 -2610.552
```

... and the corresponding logarithms of chemical activities of a set of basis species:

```
> basis(c("graphite", "H2"), c("cr", "gas"))

   C H ispecies logact state
C  1 0     1937      0    cr
H2 0 2     3090      0   gas

> basis.logact(ep)

        C        H2
 4.297937 -3.827104
```

5.2 Bulk composition instead of moles of species

run.wjd() can be given B (bulk chemical composition), instead of Y (initial numbers of moles of species). When this happens, the function tests the different initial compositions generated by guess() until one is found that is within the specified tolerance of chemical potentials of elements. Here's an example:

```
> waa <- run.wjd(c(ialk, iaro), B="C100H70")

subcrt: 8 species at 298.15 K and 1 bar
run.wjd: got within tolerance on initial solution 1 of 7

> bp <- barplot(waa$X, ylab="moles")
> text(bp, rep(0.2,8), c(alkanes, aromatics), srt=90, adj=0)
```
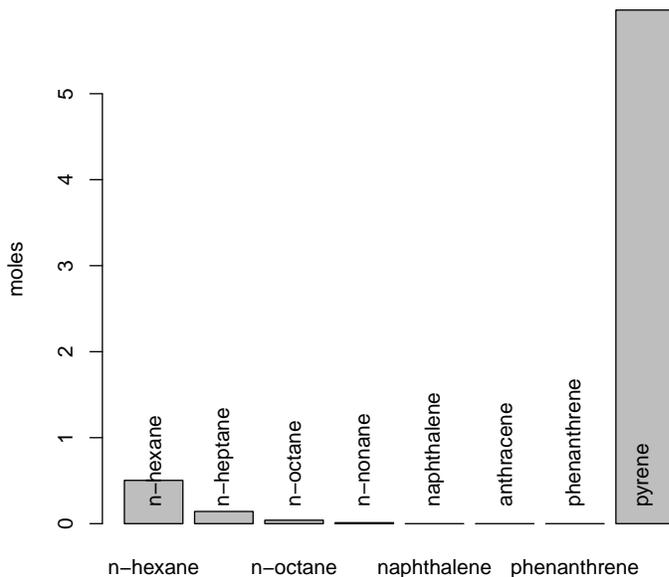

Compared with the previous example, here the lower H/C = 0.7 defines a more oxidized system, so the increase in aromatic content is expected. As before, we can calculate the chemical potentials of the elements and of a corresponding set of basis species:

```
> print(ep <- equil.potentials(waa))

        C          H
 5964.177 -2723.926

> basis.logact(ep)

        C          H2
 4.371781 -3.993311
```

6 Document Information

Revision history

- 2012-01-01 Initial version

- 2012-06-16 Running Down using alkanes and aromatics

- 2012-09-20 Add example from Dayhoff et al.

R session information

```
> sessionInfo()

R Under development (unstable) (2013-07-03 r63169)
Platform: x86_64-slackware-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US       LC_NUMERIC=C         LC_TIME=en_US
 [4] LC_COLLATE=C         LC_MONETARY=en_US    LC_MESSAGES=en_US
 [7] LC_PAPER=en_US       LC_NAME=C            LC_ADDRESS=C
[10] 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_1.0.1

loaded via a namespace (and not attached):
[1] tools_3.1.0
```

References

M. O. Dayhoff, E. R. Lippincott, and R. V. Eck. Thermodynamic Equilibria In Prebiological Atmospheres. *Science*, 146(3650):1461–1464, December 1964. doi: 10.1126/science.146.3650.1461.

M. O. Dayhoff, E. R. Lippincott, R. V. Eck, and G. Nagarajan. Thermodynamic Equilibrium in Prebiological Atmospheres of C, H, O, N, P, S, and Cl. Report SP-3040, National Aeronautics and Space Administration, Washington, D. C., 1967. URL http://ntrs.nasa.gov/search.jsp?R=19670017966.

D. K. Nordstrom, L. N. Plummer, T. M. L. Wigley, T. J. Wolery, J. W. Ball, E. A. Jenne, R. L. Bassett, D. A. Crerar, T. M. Florence, B. Fritz, M. Hoffman, Jr. Holdren, G. R., G. M. Lafon, S. V. Mattigod, R. E. McDuff, F. Morel, M. M. Reddy, G. Sposito, and J. Thrailkill. A comparison of computerized chemical models for equilibrium calculations in aqueous systems. In E. A. Jenne, editor, *Chemical Modeling in Aqueous Systems: Speciation, Sorption, Solubility, and Kinetics*, Series 93, pages 857 – 892. American Chemical Society, 1979. URL http://www.worldcat.org/oclc/4591313.

H. G. Othmer. Nonuniqueness of Equilibria In Closed Reacting Systems. *Chem. Eng. Sci.*, 31(11):993–1003, 1976. doi: 10.1016/0009-2509(76)87020-0.

W. B. White, S. M. Johnson, and G. B. Dantzig. Chemical equilibrium in complex mixtures. *J. Chem. Phys.*, 28(5):751 – 755, 1958. ISSN 0021-9606. doi: 10.1063/1.1744264.