COdos: CO2 Correction Tools

Installation

You can install the released version of codos from CRAN with:

```r
install.packages("codos")
```

And the development version from GitHub with:

```r
# install.packages("devtools")
devtools::install_github("special-uor/codos", "dev")
```

Note: Some of the equations on this document are not displayed properly (due to a server issue), check out the README.pdf.

Background:

Vapour-pressure deficit (`vpd`)

`vpd` is given by mean daily growing season temperature, `tmp` [°C] and moisture index, `mi` [-]. Using the CRU TS 4.04 dataset (University of East Anglia Climatic Research Unit et al. 2020) we found the following relation:

$$\text{vpd} = 4.589 \times \exp(0.0611 \times \text{tmp} - 0.87 \times \text{mi})$$

The steps performed were:

1. Generate a monthly climatology for the period between 1961 and 1990 (inclusive). Variables used: `cld`, `pre`, `tmn`, `tmx`, `vap`.

```r
# Monthly climatology for `tmn`
codos::monthly_clim("cru_ts4.04.1901.2019.tmn.dat.nc", "tmn", 1961, 1990)
```

Output file:

```
"cru_ts4.04.1901.2019.tmn.dat-clim-1961-1990.nc"
```

2. Interpolate the monthly data to daily. Variables used: `cld`, `pre`, `tmn`, `tmx`, `vap`.

```r
# Monthly to daily interpolation for `tmn`
codos::nc_int("cru_ts4.04.1901.2019.tmn.dat-clim-1961-1990.nc", "tmn")
```

Output file:

```
"cru_ts4.04.1901.2019.tmn.dat-clim-1961-1990-int.nc"
```

3. Calculate daily temperature, `tmp`. Variables used: `tmn` and `tmx`.

```r
codos::daily_temp(tmin = list(filename = "cru_ts4.04.1901.2019.tmn.dat-clim-1961-1990-int.nc",
                              id = "tmn"),
                  tmax = list(filename = "cru_ts4.04.1901.2019.tmx.dat-clim-1961-1990-int.nc",
                              id = "tmx"),
                  output_filename = "cru_ts4.04-clim-1961-1990-daily.tmp.nc")
```

4. Calculate mean growing season for daily temperature

```r
codos::nc_gs("cru_ts4.04-clim-1961-1990-daily.tmp.nc", "tmp", thr = 0)
```

Output file:

```
"cru_ts4.04-clim-1961-1990-daily.tmp-gs.nc"
```

5. Calculate potential evapotranspiration (`pet`)

Install `SPLASH` (unofficial R package) as follows:

```r
remotes::install_github("villegar/splash", "dev")
```

Or, download from the official source: https://bitbucket.org/labprentice/splash.

```r
elv <- codos:::nc_var_get("halfdeg.elv.nc", "elv")$data
tmp <- codos:::nc_var_get("cru_ts4.04.1901.2019.daily.tmp.nc", "tmp")$data
cld <- codos:::nc_var_get("cru_ts4.04.1901.2019.cld.dat-clim-1961-1990-int.nc",
                          "cld")$data

codos::splash_evap(output_filename = "cru_ts4.04-clim-1961-1990-pet.nc",
                   elv, # Elevation, 720x360 grid
                   sf = 1 - cld / 100,
                   tmp,
                   year = 1961, # Reference year
                   lat = codos::lat,
                   lon = codos::lon)
```

Output file:

```
"cru_ts4.04-clim-1961-1990-pet.nc"
```

6. Calculate moisture index (`mi`)

$$MI_{i,j} = \frac{\text{Total precipitation}}{\text{Total PET}}$$

```r
pet <- codos:::nc_var_get("cru_ts4.04-clim-1961-1990-pet.nc",
                          "pet")$data
pre <- codos:::nc_var_get("cru_ts4.04.1901.2019.pre.dat-new-clim-1961-1990-int.nc",
                          "pre")$data
codos::nc_mi(output_filename = "cru_ts4.04-clim-1961-1990-mi.nc",
             pet, # potential evapotranspiration
             pre) # precipitation
```

Output file:

```
"cru_ts4.04-clim-1961-1990-mi.nc"
```

7. Approximate `vpd`

```r
tmp <- codos:::nc_var_get("cru_ts4.04-clim-1961-1990-daily.tmp.nc",
                          "tmp")$data
vap <- codos:::nc_var_get("cru_ts4.04.1901.2019.vap.dat-clim-1961-1990-int.nc",
                          "vap")$data
output_filename <- file.path(path, "cru_ts4.04-clim-1961-1990-vpd-tmp.nc")
codos::nc_vpd(output_filename, tmp, vap)
```

Output file:

```
"cru_ts4.04-clim-1961-1990-vpd-tmp.nc"
```

8. Find the coeffients for the following equation

$$\text{vpd} = a \times \exp(\text{kTmp} \times \text{tmp} - \text{kMI} \times \text{mi})$$

```r
mi <- codos:::nc_var_get("cru_ts4.04-clim-1961-1990-mi.nc", "mi")$data
Tmp <- codos:::nc_var_get("cru_ts4.04-clim-1961-1990-daily.tmp-gs.nc", "tmp")$data
vpd <- codos:::nc_var_get("cru_ts4.04-clim-1961-1990-vpd-tmp-gs.nc", "vpd")$data

# Apply ice mask
mi[codos:::ice_mask] <- NA
Tmp[codos:::ice_mask] <- NA
vpd[codos:::ice_mask] <- NA

# Filter low temperatures, Tmp < 5
mi[Tmp < 5] <- NA
Tmp[Tmp < 5] <- NA

# Create data frame
df <- tibble::tibble(Tmp = c(Tmp),
                     vpd = c(vpd),
                     MI = c(mi))

# Filter grid cells with missing Tmp, vpd, or MI
df <- df[!is.na(df$Tmp) & !is.na(df$vpd) & !is.na(df$MI), ]

# Linear approximation
lmod <- lm(log(vpd) ~ Tmp + MI, data = df)
# Non-linear model
exp_mod <- nls(vpd ~ a * exp(kTmp * Tmp - kMI * MI),
               df,
               start = list(a = exp(coef(lmod)[1]),
                            kTmp = coef(lmod)[2],
                            kMI = coef(lmod)[3]),
               control = list(maxiter = 200))
```

Summary statistics:

```r
summary(exp_mod)
#>
#> Formula: vpd ~ a * exp(kTmp * Tmp - kMI * MI)
#>
#> Parameters:
#>      Estimate Std. Error t value Pr(>|t|)
#> a    4.589148   0.019843   231.3   <2e-16 ***
#> kTmp 0.061108   0.000174   351.2   <2e-16 ***
#> kMI  0.870229   0.002585   336.7   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.241 on 60291 degrees of freedom
#>
#> Number of iterations to convergence: 8
```

```r
#> Achieved convergence tolerance: 7.419e-06
coefficients(exp_mod)
#>          a       kTmp        kMI
#> 4.58914835 0.06110768 0.87022950
```

Corrected `mi` from reconstructed `mi`

The following equations were used:

$$f = e/1.6 = D/[\mathsf{c_a}(1-\chi)]$$

$$\chi = \frac{\xi \times \mathsf{vpd}^{1/2} + \mathsf{vpd}}{\mathsf{c_a}/\left(\mathsf{c_a} + 9.7\right)}$$

$$\xi = [\beta(K + \Gamma^*)/(1.6\eta^*)]^{1/2}$$

where:

- e ratio of water lost to carbon fixed [–]
- vpd vapour pressure deficit [Pa]
- c_a ambient CO2 partial pressure [Pa]
- χ ratio of leaf-internal to ambient CO2 partial pressures [–]
- ξ stomatal sensitivity factor [Pa1/2]
- Γ^* photorespiratory compensation point [Pa]: a function of temperature and elevation
- β ratio of cost factors for carboxylation and transpiration = 146 [–]
- K effective Michaelis constant of Rubisco [Pa]: a function of temperature and elevation
- η^* viscosity of water relative to its value at 25°C [–]

And the equilibrium relation:

$$f(\mathsf{T_{c1}}, \mathsf{MI_1}, \mathsf{c_{a,1}}) = f(\mathsf{T_{c0}}, \mathsf{MI_0}, \mathsf{c_{a,0}})$$

where:

- T_{c1} past temperature (assume equal to reconstructed value) [K]
- MI_1 past MI (unknown) [–]
- $c_{a,1}$ past ambient CO2 partial pressure [Pa], adjusted for elevation
- T_{c0} present temperature [K]
- MI_0 reconstructed MI [–]
- $c_{a,1}$ 'recent' ambient CO2 partial pressure [Pa], adjusted for elevation

Steps in the solution:

1. Evaluate $f(\mathsf{T_{c0}}, \mathsf{MI_0}, \mathsf{c_{a,0}})$
2. Equate this to:

$$[\xi(\mathsf{T_{c1}}, z) \times \mathsf{vpd}_1^{1/2} + \mathsf{vpd}_1]/[\mathsf{c_{a,1}}(z)/(\mathsf{c_{a,1}}(z) + 9.7)]$$

where:

- z is elevation
- vpd_1 is past vapour pressure deficit

And solve for vpd_1.

3. Convert vpd_1 back to MI (at temperature T_{c1}), to yield an estimate of MI_1.

Using `codos`, all the steps translate to a simple function call

```
corrected_mi <- codos::corrected_mi(present_t,
                                    past_temp,
                                    recon_mi,
                                    modern_co2,
                                    past_co2)
```

Note that this function takes temperatures in [°C] and ambient CO_2 partial pressures in [μmol/mol] (unless, `scale_factor` is overwritten, e.g. `scale_factor = 1` to use ambient CO_2 partial pressures in [Pa]).

More details:

```
?codos::corrected_mi
```

References

University of East Anglia Climatic Research Unit, Ian C. Harris, Philip D. Jones, and Tim Osborn. 2020. *CRU TS4.04: Climatic Research Unit (CRU) Time-Series (TS) Version 4.04 of High-Resolution Gridded Data of Month-by-Month Variation in Climate (Jan. 1901- Dec. 2019)*. Centre for Environmental Data Analysis. https://catalogue.ceda.ac.uk/uuid/89e1e34ec3554dc98594a5732622bce9.