Metabolism Estimates from Buoy Data

Here, we will use the hourly buoy data from the morning session to produce daily estimates of metabolism using the LakeMetabolizer package. https://github.com/GLEON/LakeMetabolizer (https://github.com/GLEON/LakeMetabolizer)

Set the working directory and load packages:

```
setwd("~/Dropbox/G18/")
library(readr) # read in data
library(dplyr,quietly = T) # clean data
library(tidyr) # clean data
library(rLakeAnalyzer) # lake analyses
library(lubridate) # working with time
library(LakeMetabolizer) # lake analyses
```

Read in hourly buoy and met data. Met data will shorten to select only the columns of interest.

```
buoyWide = read_csv('data/Sparkling2014wtemp_hourly_wide.csv')

met = read_csv('data/Sparkling2014domet_hourly.csv')
metData <- met %>%
  dplyr::select(sampledate,air_temp,rel_hum,wind_speed_2m,par,opt_wtemp,opt_do_raw)
```

Look at the first three rows of our data

```
head(buoyWide,3)
```

```
## Source: local data frame [3 x 29]
##
##              datetime    wtr_0 wtr_0.25  wtr_0.5 wtr_0.75    wtr_1
##                <time>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2014-05-18 12:00:00 9.437500 8.279333 9.314333 9.285833 9.266333
## 2 2014-05-18 13:00:00 9.499333 8.322833 9.408667 9.375500 9.334833
## 3 2014-05-18 14:00:00 9.922167 8.378333 9.737500 9.652000 9.544833
## Variables not shown: wtr_1.5 <dbl>, wtr_2 <dbl>, wtr_2.5 <dbl>, wtr_3
##   <dbl>, wtr_3.5 <dbl>, wtr_4 <dbl>, wtr_4.5 <dbl>, wtr_5 <dbl>, wtr_5.5
##   <dbl>, wtr_6 <dbl>, wtr_6.5 <dbl>, wtr_7 <dbl>, wtr_7.5 <dbl>, wtr_8
##   <dbl>, wtr_9 <dbl>, wtr_10 <dbl>, wtr_11 <dbl>, wtr_12 <dbl>, wtr_13
##   <dbl>, wtr_14 <dbl>, wtr_15 <dbl>, wtr_16.5 <dbl>, wtr_18 <dbl>.
```

```
head(metData,3)
```

```
## Source: local data frame [3 x 7]
##
##            sampledate air_temp  rel_hum wind_speed_2m      par opt_wtemp
##                <time>    <dbl>    <dbl>         <dbl>    <dbl>     <dbl>
## 1 2014-05-18 12:00:00 18.90333 25.23333      2.811667 986.7500  10.56383
## 2 2014-05-18 13:00:00 17.85667 28.56667      3.353333 582.7233  10.58000
## 3 2014-05-18 14:00:00 18.67167 27.31500      2.473333 811.6083  10.89800
## Variables not shown: opt_do_raw <dbl>.
```

What is needed for metabolism estimates?

Look in the LakeMetabolizer help files using `?metab`

1. date vector
2. DO observations
3. DO saturation
4. mixing depth (z.mix)
5. PAR values (day vs. night for book keeping method)
6. water temperatures
7. gas exchange (k.gas)

1-3:

Set date vector, DO observations, and DO saturation

```
datetime = metData$sampledate
do.obs = metData$opt_do_raw
# Get equilibrium saturation concentration of oxygen in water
do.sat = o2.at.sat.base(metData$opt_wtemp, altitude=300)
```

4: z.mix

For mixing depth, we will use a rLakeAnalyzer function

```
# Z mix
z.mix = ts.thermo.depth(wtr = buoyWide,na.rm = T)[,2]
```

5: PAR values

We have PAR values, but we also need to calculate boolean values for day vs. night.
Where 1 = day, and 0 = night.

```
par = metData$par
# For bookeeping irradiance
isday = is.day(metData$sampledate,48)
dayIrr = rep(0,nrow(metData))
dayIrr[isday] = 1
```

6: Water temperatures

Water temperatures should be taken at same depth as DO sensor.
In this case, 0.5 m

```
wtr = buoyWide$wtr_0.5
```

7: k.gas

What is needed for k.gas? Look in LakeMetabolizer help.

```
?k.cole.base
k.cole.base(wnd)
?k.read.base
k.read.base(wnd.z, Kd, lat, lake.area, atm.press, dateTime, Ts, z.aml, airT, wnd, RH, sw, lwnet)
```

First, scale wind speed to standard U10 (10 meters) based on height of observation

```
U10 <- wind.scale.base(wnd = metData$wind_speed_2m, wnd.z = 2)
```

For k.read, we need shortwave radiation. Luckily, we have PAR.
Also need longwave radiation

```
# Calculate shortwave radiation
sw = par.to.sw.base(metData$par)

# Calculate longwave radiation
airtemp = metData$air_temp
rh = metData$rel_hum
lwnet = calc.lw.net.base(datetime,sw,wtr,48,1013,airtemp,rh)
```

Calculate k600 values

```
# k600 Cole
k600.cole = k.cole.base(U10)
# k600 Read
k600.read = k.read.base(10,2,48,5000,1013,datetime,wtr,z.mix,airtemp,
                        U10,rh,sw,lwnet)
```

```
## Warning in sw * 0.93 - E: longer object length is not a multiple of shorter
## object length
```

```
## Warning in sw * 0.93 - E - H: longer object length is not a multiple of
## shorter object length
```

```
## Warning in C_D * wnd^2: longer object length is not a multiple of shorter
## object length
```

Convert k600 values to gas flux

```
# k.gas flux
kgas.cole = k600.2.kGAS.base(k600.cole, wtr, 'O2')
kgas.read = k600.2.kGAS.base(k600.read, wtr, 'O2')
```

Calculate metabolism using multiple methods

For example: Using metab.mle

```
metab.mle(do.obs, do.sat, kgas.cole, z.mix, par, wtr)
```

```
## $params
##      gppCoeff        rCoeff             Q           nll        doInit
## -3.414059e-05  4.215662e-03  3.218157e-02  4.590945e+03  1.127510e+01
##
## $metab
##       GPP         R       NEP
## -33.48838  32.11307  -1.37531
```

These results are metabolism for the entire period of our data. Instead, we want to calculate metabolism each day, and output NEP.

Since we have two estimates of gas flux **kgas.cole** and **kgas.read**, write a function that can accept any gas flux as an argument, and output a data.frame of metabolism estimates.

```
metab.estimates <- function(kgas) {
  # Setup empty data.frame for three months
  output = data.frame(date = seq.Date(as.Date('2014-06-01'),as.Date('2014-08-31'),by='da
y'),NEP.mle = NA, NEP.ols=NA, NEP.kalman = NA, NEP.bookkeep = NA)

  # Loop through each day.
  for (i in 1:nrow(output)) {
    # Create a index for each day, so only those days values will be used
    indx <- as.Date(metData$sampledate) == output$date[i]

    m.mle = metab.mle(do.obs[indx], do.sat[indx], kgas[indx], z.mix[indx],
par[indx],wtr[indx])
    m.ols = metab.ols(do.obs[indx], do.sat[indx], kgas[indx], z.mix[indx],
par[indx],wtr[indx])
    m.kal = metab.kalman(do.obs[indx], do.sat[indx], kgas[indx], z.mix[indx], par[indx],w
tr[indx])
    m.book = metab.bookkeep(do.obs[indx], do.sat[indx], kgas[indx], z.mix[indx], dayIrr[i
ndx])

    output$NEP.mle[i] = round(m.mle$metab[3],3)
    output$NEP.ols[i] = round(m.ols$metab[3],3)
    output$NEP.kalman[i] = round(m.kal$metab[3],3)
    output$NEP.bookkeep[i] = round(m.book$NEP,3)
  }
  return(output)
}
```

Run function

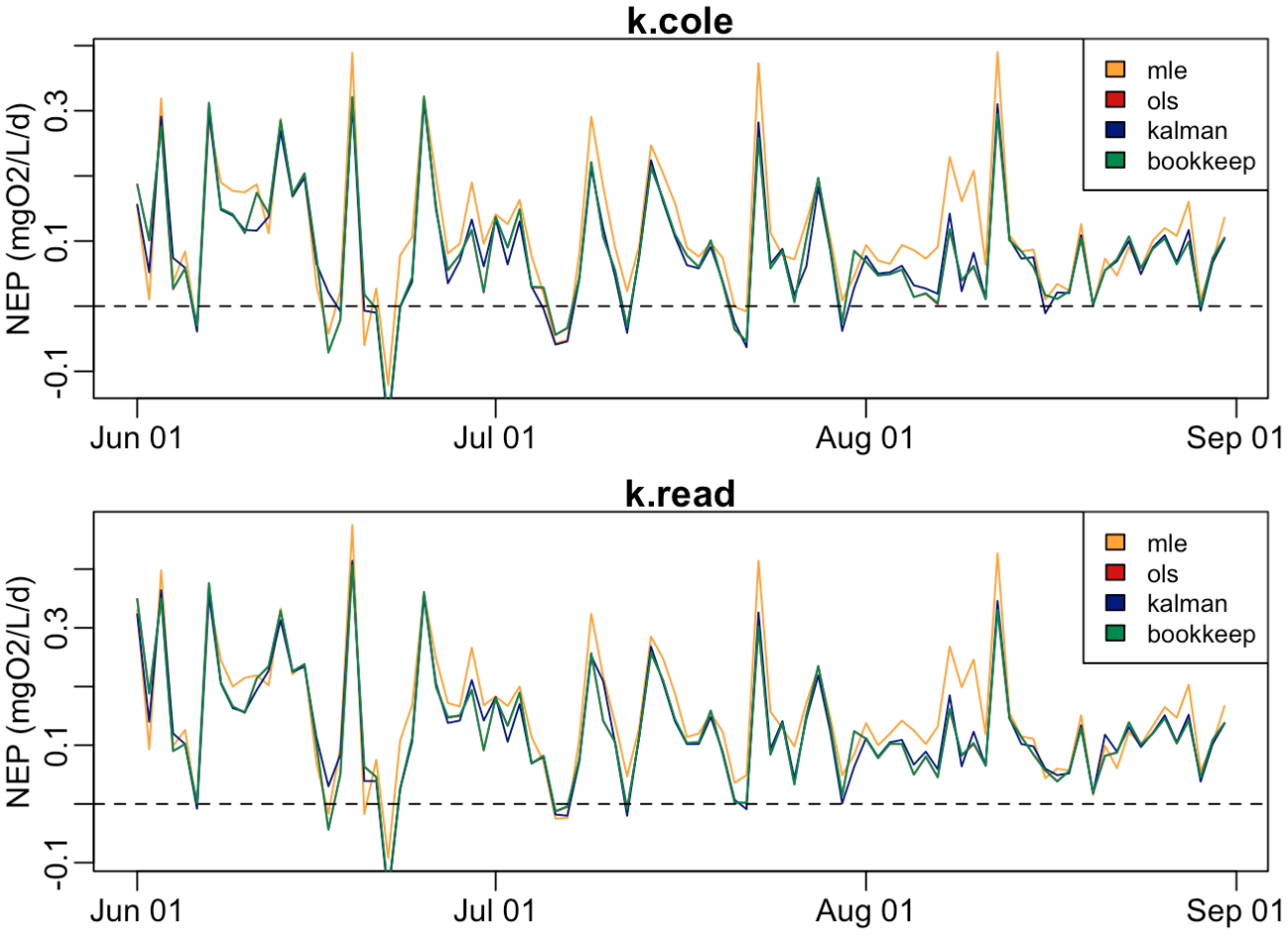
```
metab.cole = metab.estimates(kgas.cole)
metab.read = metab.estimates(kgas.read)
```

Plot results

```
par(mfrow=c(2,1),mar=c(2,3,1,1),mgp=c(1.5,0.5,0))

plot(metab.cole$date,metab.cole$NEP.mle,type='l',col='orange',
     ylab='NEP (mgO2/L/d)',xlab='',main='k.cole')
lines(metab.cole$date,metab.cole$NEP.ols,col='red3')
lines(metab.cole$date,metab.cole$NEP.kalman,col='navy')
lines(metab.cole$date,metab.cole$NEP.bookkeep,col='springgreen4')
legend('topright',legend = c('mle','ols','kalman','bookkeep'),
       fill = c('orange','red3','navy','springgreen4'),cex=0.8)
abline(h=0,lty=2)


plot(metab.read$date,metab.read$NEP.mle,type='l',col='orange',
     ylab='NEP (mgO2/L/d)',xlab='',main='k.read')
lines(metab.read$date,metab.read$NEP.ols,col='red3')
lines(metab.read$date,metab.read$NEP.kalman,col='navy')
lines(metab.read$date,metab.read$NEP.bookkeep,col='springgreen4')
legend('topright',legend = c('mle','ols','kalman','bookkeep'),
       fill = c('orange','red3','navy','springgreen4'),cex=0.8)
abline(h=0,lty=2)
```



Output final data.frame

Here, we select just the two MLE estimates.

```
final = data.frame(datetime = metab.cole$date, NEP.mle.cole = metab.cole$NEP.mle, NEP.ml
e.read = metab.read$NEP.mle)
write.csv(final,'data/Sparkling2014_NEP.csv',row.names=F)
```