```
##############################################################################
# NRC HICU-1
# Isotope ratio measurements of copper using gravimetric isotope mixture model
#
# Purity GDMS data: Brad Methven (Jan 2020)
# Isotope ratio MCICPMS data: Lu Yang (Dec 2020 to Mar 2021)
# Final data analysis: Juris Meija (Dec 2021, rev. Jan 2023, rev. Jun 2023)
##############################################################################

dev.off()

# Read the data

require(readxl)
setwd('C:/Users/meijaj/Documents/jm-NRC/NRC_CRM/HICU-1/HICU-1-isotope-ratio')

df_meas      = data.frame(read_xlsx('FGIM-Cu-Dec2021.xlsx', sheet = '1_measurements',
col_types = c('text','text','numeric','numeric','text','text',rep('numeric',11))))
df_mixtures  = data.frame(read_xlsx('FGIM-Cu-Dec2021.xlsx', sheet = '2_mixtures', col_types
= c('text','text','text',rep('numeric',6))))
df_solutions = data.frame(read_xlsx('FGIM-Cu-Dec2021.xlsx', sheet = '3_solutions',
col_types = c('text','text',rep('numeric',6))))
df_stock     = data.frame(read_xlsx('FGIM-Cu-Dec2021.xlsx', sheet = '4_stock', col_types =
c('text','text',rep('numeric',4))))

# Purity of copper isotope materials
pA_meas = df_stock[df_stock$'sample_id'=='A',c('purity_g_g', 'u_purity_g_g')]
pB_meas = df_stock[df_stock$'sample_id'=='B',c('purity_g_g', 'u_purity_g_g')]

# Mass fraction of copper isotope materials in the primary stock solution
wA_stock = df_stock[df_stock$'sample_id'=='A',c('w_stock_mg_kg', 'u_w_stock_mg_kg')]
wB_stock = df_stock[df_stock$'sample_id'=='B',c('w_stock_mg_kg', 'u_w_stock_mg_kg')]

# Atomic masses of copper
require(CIAAWconsensus)
m_a_Cu = unlist(subset(ciaaw.mass.2016, isotope %in% c('63Cu','65Cu'), select = 'mass'))
u_m_a_Cu = unlist(subset(ciaaw.mass.2016, isotope %in% c('63Cu','65Cu'), select =
'uncertainty'))

# Gather mass data for gravimetric mixtures
m_data_id = grepl('AB', df_mixtures$sample_id)
m_data = df_mixtures[m_data_id, c('set_id','sample_id', 'mass_A_g', 'mass_B_g', 'u_m_g')]
m_N = length(m_data$mass_A_g)

# Produce a random draw for the mass fraction of copper in
# solutions A and B for each set consistent with measurement uncertainties
get_wAB = function(k=1){
  # Purity of copper isotope materials (truncated to be <= 1 g/g)
  pA_it = min(1, rnorm(1, pA_meas[1,'purity_g_g'], k * pA_meas[1,'u_purity_g_g']))
  pB_it = min(1, rnorm(1, pB_meas[1,'purity_g_g'], k * pB_meas[1,'u_purity_g_g']))

  # Mass fraction of copper isotope materials in the primary stock solution
  wA_it = rnorm(1, wA_stock[1,'w_stock_mg_kg'], k * wA_stock[1,'u_w_stock_mg_kg'])
  wB_it = rnorm(1, wB_stock[1,'w_stock_mg_kg'], k * wB_stock[1,'u_w_stock_mg_kg'])

  # Mass of copper in the stock solution
  wCu_stock_it = c('A'=pA_it * wA_it, 'B'=pB_it * wB_it)
```

```
  # Dilution of the stock solutions
  N = length(df_solutions[,'m1_g'])
  m1_it = rnorm(N, df_solutions[,'m1_g'], df_solutions[,'u_m_g'])
  m2_it = rnorm(N, df_solutions[,'m2_g'], df_solutions[,'u_m_g'])
  m1final_it = rnorm(N, df_solutions[,'m1_final_g'], df_solutions[,'u_m_g'])
  m2final_it = rnorm(N, df_solutions[,'m2_final_g'], df_solutions[,'u_m_g'])

  id = match(names(wCu_stock_it), df_solutions[,'sample_id'])

  res=data.frame(
          'set'=df_solutions[,'set_id'],
          'sample'=df_solutions[,'sample_id'],
          'wCu'=(m1_it/m1final_it)*(m2_it/m2final_it)*wCu_stock_it[id]
          )
  return(res)
}

# Calculation steps for set_id and analysis_id
# 1 Construct polynomial drift function for 65/63 isotope ratio from samples AB1-x as a
function of measurement_id
# 2 Apply drift function to all observed isotope ratios
# 3 For each gravimetric mix sample (AB1-1 and AB1b for set 1 and AB1-1,AB2, AB3 for set2)
calculate the mass bias factor
# 4 Calculate the correction factors for the sample and obtain the corrected isotope ratio

get_R = function(set_id, analysis_id, sample_id, w, m, m_a, draw=TRUE, k=1, fit=TRUE){

  wAB = get_wAB(k=k)
  m_a = rnorm(2, m_a_Cu, k*u_m_a_Cu)
  m_data$mass_A_g_it = rnorm(m_N, m_data$mass_A_g, k*m_data$u_m_g)
  m_data$mass_B_g_it = rnorm(m_N, m_data$mass_B_g, k*m_data$u_m_g)

  #set_id='set-1'
  #analysis_id ='1'
  #sample_id='HICU-1'
  #sample_id='NIST976'
  w = wAB
  m = m_data
  m_a = m_a

  # get masses of the isotope solutions to make blends
  mAB  = m[m$set_id==set_id, c('sample_id','mass_A_g_it','mass_B_g_it')]
  mAB$mass_A_g_it = mAB$mass_A_g_it * w[w$set==set_id & w$sample=='A','wCu']
  mAB$mass_B_g_it = mAB$mass_B_g_it * w[w$set==set_id & w$sample=='B','wCu']

  # get isotope ratios of the isotope blends
  r_data_id = df_meas$set_id == set_id & df_meas$analysis_id == analysis_id &
!grepl('blank|HIZN', df_meas$sample_id)
  r_data = df_meas[r_data_id, c('measurement_id','sample_id','X63Cu.65Cu','u_63Cu.65Cu')]
  r_data$X63Cu.65Cu_it = rnorm(length(r_data$X63Cu.65Cu), r_data$X63Cu.65Cu,
r_data$u_63Cu.65Cu)

  # cubic/quadratic drift correction
  spline_data = r_data[grepl('AB1-', r_data$sample_id),
c('measurement_id','X63Cu.65Cu_it')]

  # eliminate outliers (Aug 2022) JM
  if(set_id=='set-1'&analysis_id=='1') spline_data = spline_data[-2,]
```

```
  if(set_id=='set-1'&analysis_id=='4') spline_data = spline_data[-3,]

  # correction (Aug 2022) JM
  spline_data[,2] = spline_data[,2]/spline_data[1,2]
  sp = lm(X63Cu.65Cu_it ~ poly(measurement_id, 2), spline_data)

  if(draw){
    plot(x=spline_data$measurement_id, y=spline_data$X63Cu.65Cu_it)
    p = predict(sp, newdata = data.frame(measurement_id=1:25))
    lines(x=1:25, y=p)
  }

  # apply drift correction for isotope ratios with AB1-1 set to zero-drift
  r_data$drift_corr = predict(sp, newdata = list(measurement_id = r_data$measurement_id))
  r_data$X63Cu.65Cu_drift_corr = r_data$X63Cu.65Cu_it * r_data$drift_corr

  # calculate the isotope ratio correction factor for each grav. mix. sample AB
  ab_id = grepl('AB', r_data$sample_id)
  rA  = r_data[r_data$sample_id=='A', 'X63Cu.65Cu_drift_corr']
  rB  = r_data[r_data$sample_id=='B', 'X63Cu.65Cu_drift_corr']
  rAB = r_data[ab_id, c('sample_id','X63Cu.65Cu_drift_corr')]

  # merge mass data and the isotope ratio data
  dK = merge(mAB, rAB)
  names(dK) = c('id','mA','mB','rAB')

  # calculate isotope ratio correction factor for each mixture
  dK$K = (m_a[2]/m_a[1])*(dK$mA*(dK$rAB - rA) + dK$mB*(dK$rAB - rB))/(dK$mA*rB*(rA -
dK$rAB) + dK$mB*rA*(rB - dK$rAB))

  # Average K factor for r = 1
  K = mean(dK[dK$rAB < 1.05 & dK$rAB > 0.95,'K'])

  # calculate isotope ratios for the sample
  s1 = which(r_data$sample_id == sample_id)
  r_sample = r_data[s1, 'X63Cu.65Cu_drift_corr']

  drift = r_data$drift_corr

  if(fit) return ( K * r_sample )
  if(!fit) return ( list(dK, K * r_sample, r_sample, rA, rB, drift, spline_data))
}


# FIGURE 0 # Isotope ratio drifts during the measurement sequences

plot(x=df_meas$measurement_id, y=df_meas$drift_Cu, ylim=c(0.998,1.002),pch='.')
text(x=df_meas$measurement_id, y=df_meas$drift_Cu, labels = df_meas$analysis_id)
# Set 1 Sequence 3 shows excessive isotope ratio drift

# END FIGURE 0

# FIGURE 1 # Manuscript (4.6 x 7 in)

K_63_65_set1_1 = replicate(1e3, {
     m_a = rnorm(2, m_a_Cu, u_m_a_Cu)
     m_data$mass_A_g_it = rnorm(m_N, m_data$mass_A_g, m_data$u_m_g)
     m_data$mass_B_g_it = rnorm(m_N, m_data$mass_B_g, m_data$u_m_g)
```

```
     get_R('set-1', 1, 'HICU-1', w = wAB, m = m_data, m_a = m_a, fit=F, draw=F)[[1]]$K
   })
K_63_65_set1_2 = replicate(1e3, {
     m_a = rnorm(2, m_a_Cu, u_m_a_Cu)
     m_data$mass_A_g_it = rnorm(m_N, m_data$mass_A_g, m_data$u_m_g)
     m_data$mass_B_g_it = rnorm(m_N, m_data$mass_B_g, m_data$u_m_g)
     get_R('set-1', 2, 'HICU-1', w = wAB, m = m_data, m_a = m_a, fit=F, draw=F)[[1]]$K
   })

xx_r1=get_R('set-1', 1, 'HICU-1', w = wAB, m = m_data, m_a = m_a, fit=F, draw=F)[[1]]$rAB
xx_r2=get_R('set-1', 2, 'HICU-1', w = wAB, m = m_data, m_a = m_a, fit=F, draw=F)[[1]]$rAB

yy_K1=apply(K_63_65_set1_1,1,mean); yy_uK1=apply(K_63_65_set1_1,1,sd)
yy_K2=apply(K_63_65_set1_2,1,mean); yy_uK2=apply(K_63_65_set1_2,1,sd)

par(mar=c(5,5,1,1), mfrow=c(1,2), pty='s')
plot(x=xx_r1, y=yy_K1, log='x', xlim=c(0.1,10), ylim=c(1.055,1.0575),
     pch='', xlab=expression(''^'63'*'Cu/'^'65'*'Cu isotope ratio'),
ylab=expression(italic('K')*'('^'63'*'Cu/'^'65'*'Cu)') )
abline(v=c(1,2.24), lty=2)
segments(xx_r1, yy_K1-2*yy_uK1, xx_r1, yy_K1+2*yy_uK1, lwd=3, col=2)
points(x=xx_r1, y=yy_K1, pch=21, bg='white', cex=1.2)
mtext(side=3,line=0.4,text='95% CI', cex=0.8, adj=0.95)

plot(x=xx_r2, y=yy_K2, log='x', xlim=c(0.1,10), ylim=c(1.0515,1.0545),
     pch=19, xlab=expression(''^'63'*'Cu/'^'65'*'Cu isotope ratio'),
ylab=expression(italic('K')*'('^'63'*'Cu/'^'65'*'Cu)') )
abline(v=c(1,2.24), lty=2)
segments(xx_r2, yy_K2-2*yy_uK2, xx_r2, yy_K2+2*yy_uK2, lwd=3, col=2)
points(x=xx_r2, y=yy_K2, pch=21, bg='white', cex=1.2)
mtext(side=3,line=0.4,text='95% CI', cex=0.8, adj=0.95)

# END FIGURE 1

# Isotopic abundances and atomic weight of copper in NRC HICU-1
# 63Cu/65CU isotope ratio

# individual values from each measurement sequence
R_hicu1_individual = replicate(1e4, {
  wAB = get_wAB()
  m_a = rnorm(2, m_a_Cu, u_m_a_Cu)
  m_data$mass_A_g_it = rnorm(m_N, m_data$mass_A_g, m_data$u_m_g)
  m_data$mass_B_g_it = rnorm(m_N, m_data$mass_B_g, m_data$u_m_g)
  r1 = sapply(c(1:5), function(analysis_id) get_R('set-1', analysis_id, 'HICU-1', w = wAB,
m = m_data, m_a = m_a, draw=F))
  r2 = sapply(6:9, function(analysis_id) get_R('set-2', analysis_id, 'HICU-1', w = wAB, m =
m_data, m_a = m_a, draw=F))
  c(r1,r2)
})

R_hicu_individual_mu = apply(R_hicu1_individual,1,mean)
R_hicu_individual_u = apply(R_hicu1_individual,1,sd)

# FIGURE 2 MANUSCRIPT (4.6 x 4.6 in)

par(mfrow=c(1,1), mar=c(5,5,1,2), pty='s')
plot(x=1:9,y=R_hicu_individual_mu, ylim=c(2.240, 2.2426),
```

```
     pch='', xlab='Measurement sequence (Dec 2020 to Mar 2021)',
ylab=expression(italic(R)[63/65]*' in HICU-1'))
segments(1:9, R_hicu_individual_mu - 2*R_hicu_individual_u,
         1:9, R_hicu_individual_mu + 2*R_hicu_individual_u, lwd=3, col=2)
points(1:9, R_hicu_individual_mu, pch=21, bg=c(1,1,'white',1,1,1,1,1,1), cex=1.3)
axis(1, at=1:9,labels=1:9)
mtext(side=3,line=0.4,text='95% CI', cex=0.8, adj=0.95)

# END FIGURE 2

# Certified (combined) values for HICU-1 ###################################
# Set 1 Seq 3 removed

 R_hicu1 = c(R_hicu1_individual[-3,])

 # Isotopic abundances and atomic weight
 x63 = R_hicu1/(1 + R_hicu1)
 x65 = 1 - x63
 m63 = rnorm(length(x63), m_a_Cu[1], u_m_a_Cu[1])
 m65 = rnorm(length(x65), m_a_Cu[2], u_m_a_Cu[2])
 ACu = x63 * m63 + x65 * m65

 # print summary
 formatC(c( mean(ACu), sd(ACu), 2*sd(ACu) ), digits=5, format='f')
 formatC(c( mean(R_hicu1),   sd(R_hicu1),   2*sd(R_hicu1)), digits=4, format='f')
 formatC(c( mean(1/R_hicu1), sd(1/R_hicu1), 2*sd(1/R_hicu1)), digits=5, format='f')
 formatC(c( mean(x63), sd(x63), 2*sd(x63)), digits=5, format='f')
 formatC(c( mean(x65), sd(x65), 2*sd(x65)), digits=5, format='f')

 ### Isotopic composition of copper in NRC HICU-1 ##############################
 #                                                                             #
 # Atomic weight, A_r(Cu)         # 63.546 07 (u = 0.000 09,  U_95 = 0.000 18)  #
 # Molar mass, M(Cu)              # 63.546 07 (u = 0.000 09,  U_95 = 0.000 18)  #
 # Isotope ratio, n(63Cu)/n(65Cu) # 2.2413    (u = 0.00045,   U_95 = 0.0009)    #
 # Isotope ratio, n(65Cu)/n(63Cu) # 0.446 17  (u = 0.000 09,  U_95 = 0.000 18)  #
 # Isotopic abundance, x(63Cu)    # 0.691 48  (u = 0.000 045, U_95 = 0.000 09)  #
 # Isotopic abundance, x(65Cu)    # 0.308 52  (u = 0.000 045, U_95 = 0.000 09)  #
 #                                                                             #
 ###############################################################################

 ### Isotopic composition of copper in NIST SRM976 #############################
 ### Derived from the expression: delta(HICU-1 vs SRM976) = R(HICU-1)/RSRM976 - 1

 R976 = rnorm(1e4, 2.2413, 0.0009/2) / (1 + rnorm(1e4, -0.04, 0.03/2)/1000)

 round(c(mean(R976), 2*sd(R976)),4)
 # 2.2414 +/- 0.0009

 ### END # 27 Jun 2023
```