Assessing 14C blanks in the small-scale analysis of *n*-Alkane compound-specific-radiocarbon-analysis

Kristina Reetz1, Ronny Friedrich2, Jago J. Birk1, 3, Wilfried Rosendahl2,4, Sabine Fiedler1

1 Johannes Gutenberg-University, Institute of Geography, Johann-Joachim-Becher Weg 21, 55128 Mainz, Germany

2 Curt- Engelhorn- Center Archaeometry, D6,3, 68159 Mannheim

3 recent Georg-August-Universität Göttingen, Institute of Geography, Goldschmidtstraße. 5, 37077 Göttingen

4 Reiss-Engelhorn-Museen, 68159 Mannheim

19.03.2022

The R-script is based on the R-script "Bayesian Blank-Assessment for Compound-Specific Radiocarbon Analysis" written by "Andrew M. Dolman" in "20 April 2020", published in: Sun, S., Meyer,V.D., Dolman, A.M. et al.(2020). 14C Blank Assessment in Small-Scale Compound-Specific Radiocarbon Analysis of Lipid Biomarkers and Lignin Phenols. Radiocarbon, 62, 207-218. doi.org/10.1017/rdc.2019.108

To run this script, you will need the following packages:

library(ggplot2)

library(dplyr)  
library(tidyr)  
library(rstan)  
library(egg)

# Alkane data

We first analyze the n-alkane data.

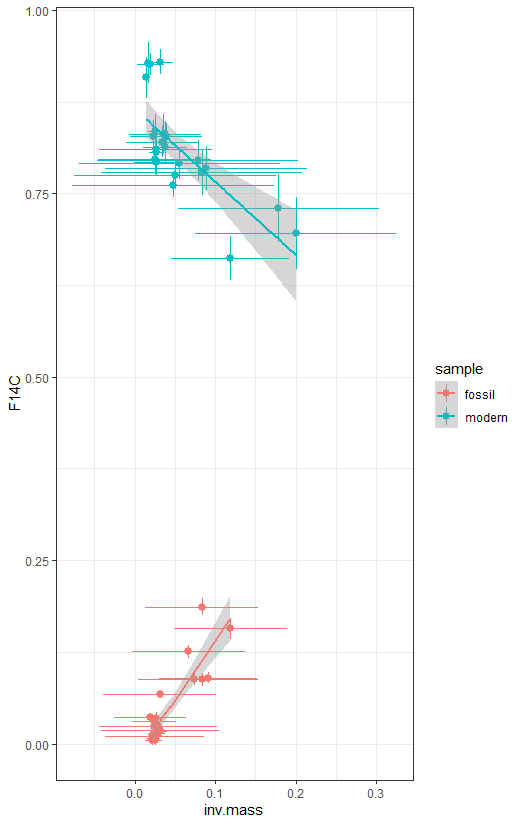
### Load the data

alkane <- read.csv("C:/Users/krreetz/Documents/R/nAlkane\_data.csv") #change the path  
alkane

### Plot the data to check it loaded correctly

This is a helper function to plot the data

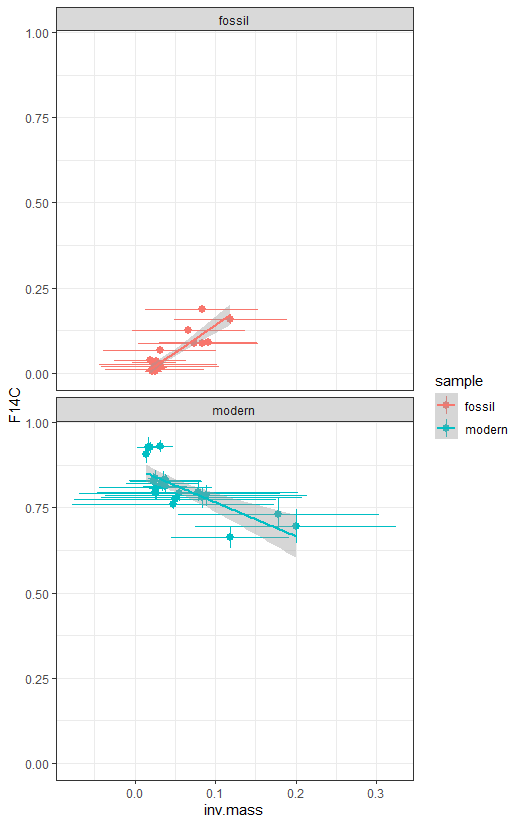
PlotData <- function(data){  
 p <- ggplot(data=data,  
 aes(x = inv.mass, y = F14C, colour = sample)) +   
 geom\_pointrange(aes(ymax = F14C + 2\*F14C.err, ymin = F14C - 2\*F14C.err)) +  
 geom\_segment(aes(xend = inv.mass + 2\*inv.mass.err,  
 x = inv.mass - 2\*inv.mass.err,  
 y = F14C, yend = F14C))+   
 geom\_smooth(method = "lm", se = T) +   
 theme\_bw() +   
 expand\_limits(x = 0)  
 p  
}

p.alk <- PlotData(alkane)   
p.alk

## `geom\_smooth()` using formula 'y ~ x'

p.alk + facet\_wrap(~sample, scales = "fixed", 0, 1)

## Warning: `nrow` is missing or less than 1 and will be treated as NULL.

## `geom\_smooth()` using formula 'y ~ x'

### 

### Fit Bayesian regression models

Bayesian modelling will allow an error model for both response and predictor variables. Posterior samples of regression parameters will allow for easy numerical estimation of distribution of intersection of the two regression lines.

The data and parameter values for a Stan model needs to be in a named list. This is helper function to create that list from the dataframe of n-alkane data.

# Function to arrange data for Stan model  
GetStanData <- function(df,  
 mu\_int\_modern = 0, sigma\_int\_modern = 10,  
 mu\_int\_fossil = 0, sigma\_int\_fossil = 10){  
   
 foss <- dplyr::filter(df, sample == "fossil")  
 mod <- dplyr::filter(df, sample == "modern")  
   
 stan.dat <- list(  
 x\_meas\_modern = mod$inv.mass,  
 x\_meas\_fossil = foss$inv.mass,  
   
 x\_err\_modern = mod$inv.mass.err,  
 x\_err\_fossil = foss$inv.mass.err,  
   
 y\_err\_modern = mod$F14C.err,  
 y\_err\_fossil = foss$F14C.err,  
   
 y\_meas\_modern = mod$F14C,  
 y\_meas\_fossil = foss$F14C,  
   
 N\_modern = length(mod$inv.mass),  
 N\_fossil = length(foss$inv.mass),  
   
 mu\_int\_modern = mu\_int\_modern,  
 sigma\_int\_modern = sigma\_int\_modern,  
   
 mu\_int\_fossil = mu\_int\_fossil,  
 sigma\_int\_fossil = sigma\_int\_fossil  
 )  
 return(stan.dat)  
 }

Here we create the list with the data and parameters for the prior distributions for the intercepts and slopes.

# create the data object  
stan.dat.alk <- GetStanData(alkane,  
 mu\_int\_modern = 0.9948,   
 sigma\_int\_modern = 0.0030,  
 mu\_int\_fossil = 0.0001,   
 sigma\_int\_fossil = 0.0001)

The Stan model is defined in a separate text file.

mod\_code <- readLines("C:/Users/krreetz/Documents/R/blank-assessment.Stan") #change the path

# Set seed so that results are replicable (same set of random numbers used)  
set.seed(20180328)  
  
# compile and fit the model  
stan.fit.alk <- rstan::stan(model\_code = mod\_code,  
 data = stan.dat.alk, iter = 3500,  
 cores = 4, chains = 3, init = "random")

Summary of the fitted model. The Rhat values should be close to 1.

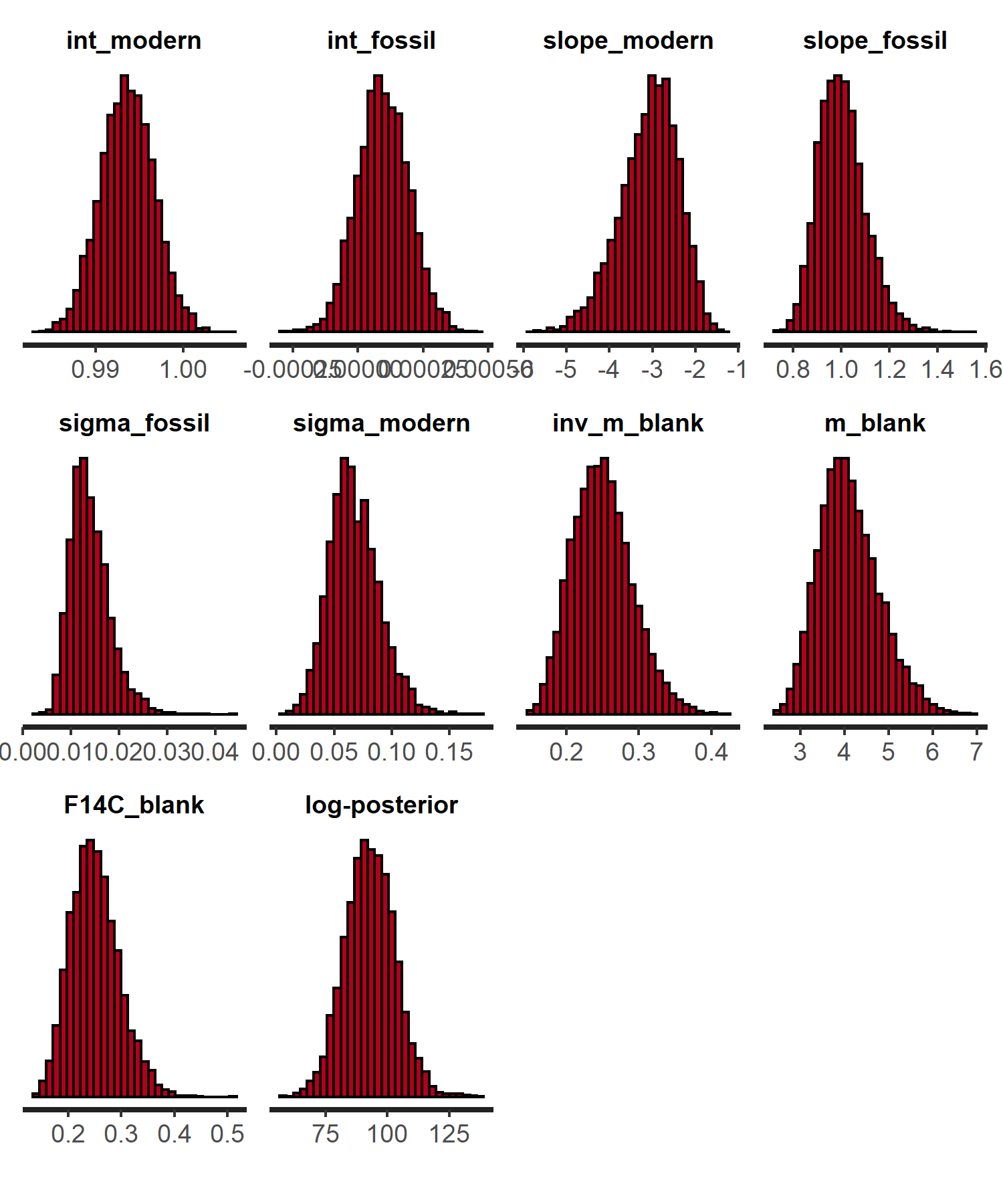
print(stan.fit.alk)

## Inference for Stan model: 00f741f5e3c8059347dec57e5056b2ba.  
## 3 chains, each with iter=3500; warmup=1750; thin=1;   
## post-warmup draws per chain=1750, total post-warmup draws=5250.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  
## x\_modern[1] 0.11 0.00 0.04 0.06 0.09 0.11 0.14 0.20 804 1  
## x\_modern[2] 0.10 0.00 0.03 0.05 0.08 0.10 0.12 0.18 941 1  
## x\_modern[3] 0.11 0.00 0.03 0.07 0.09 0.11 0.13 0.17 996 1  
## x\_modern[4] 0.07 0.00 0.03 0.03 0.05 0.07 0.09 0.13 1557 1  
## x\_modern[5] 0.07 0.00 0.03 0.03 0.05 0.07 0.09 0.14 1654 1  
## x\_modern[6] 0.07 0.00 0.03 0.02 0.05 0.06 0.08 0.13 1622 1  
## x\_modern[7] 0.07 0.00 0.03 0.02 0.05 0.06 0.08 0.13 2012 1  
## x\_modern[8] 0.07 0.00 0.03 0.02 0.05 0.07 0.08 0.13 2317 1  
## x\_modern[9] 0.07 0.00 0.03 0.03 0.06 0.07 0.09 0.13 2548 1  
## x\_modern[10] 0.05 0.00 0.02 0.02 0.04 0.05 0.06 0.08 5021 1  
## x\_modern[11] 0.04 0.00 0.01 0.02 0.04 0.04 0.05 0.07 5989 1  
## x\_modern[12] 0.04 0.00 0.02 0.01 0.03 0.04 0.06 0.08 4586 1  
## x\_modern[13] 0.04 0.00 0.00 0.03 0.03 0.04 0.04 0.04 8810 1  
## x\_modern[14] 0.04 0.00 0.02 0.01 0.03 0.04 0.05 0.08 5680 1  
## x\_modern[15] 0.03 0.00 0.01 0.01 0.02 0.03 0.03 0.05 2479 1  
## x\_modern[16] 0.05 0.00 0.02 0.01 0.04 0.05 0.06 0.09 4700 1  
## x\_modern[17] 0.03 0.00 0.00 0.02 0.02 0.03 0.03 0.04 4124 1  
## x\_modern[18] 0.04 0.00 0.01 0.01 0.03 0.04 0.04 0.06 4538 1  
## x\_modern[19] 0.05 0.00 0.02 0.01 0.04 0.05 0.06 0.09 4928 1  
## x\_modern[20] 0.03 0.00 0.00 0.02 0.02 0.03 0.03 0.03 5494 1  
## x\_modern[21] 0.02 0.00 0.00 0.02 0.02 0.02 0.03 0.03 5746 1  
## x\_modern[22] 0.02 0.00 0.01 0.01 0.01 0.02 0.02 0.03 6275 1  
## x\_modern[23] 0.02 0.00 0.00 0.01 0.02 0.02 0.02 0.02 8627 1  
## x\_modern[24] 0.01 0.00 0.00 0.01 0.01 0.01 0.02 0.02 8910 1  
## x\_fossil[1] 0.15 0.00 0.02 0.11 0.14 0.15 0.16 0.19 2487 1  
## x\_fossil[2] 0.09 0.00 0.01 0.06 0.08 0.09 0.10 0.12 3766 1  
## x\_fossil[3] 0.17 0.00 0.02 0.13 0.16 0.17 0.19 0.21 2076 1  
## x\_fossil[4] 0.09 0.00 0.02 0.06 0.08 0.09 0.10 0.12 2810 1  
## x\_fossil[5] 0.09 0.00 0.02 0.06 0.08 0.09 0.10 0.12 3411 1  
## x\_fossil[6] 0.12 0.00 0.02 0.08 0.11 0.12 0.13 0.15 2445 1  
## x\_fossil[7] 0.02 0.00 0.01 -0.01 0.01 0.02 0.03 0.05 4684 1  
## x\_fossil[8] 0.06 0.00 0.01 0.03 0.05 0.06 0.07 0.09 3152 1  
## x\_fossil[9] 0.03 0.00 0.00 0.02 0.03 0.03 0.03 0.04 6780 1  
## x\_fossil[10] 0.02 0.00 0.01 0.00 0.02 0.03 0.03 0.05 5367 1  
## x\_fossil[11] 0.03 0.00 0.00 0.02 0.02 0.03 0.03 0.04 6183 1  
## x\_fossil[12] 0.03 0.00 0.01 0.00 0.02 0.03 0.03 0.05 4872 1  
## x\_fossil[13] 0.03 0.00 0.00 0.02 0.02 0.03 0.03 0.03 8089 1  
## x\_fossil[14] 0.03 0.00 0.00 0.02 0.03 0.03 0.03 0.04 7607 1  
## x\_fossil[15] 0.02 0.00 0.00 0.02 0.02 0.02 0.03 0.03 6613 1  
## x\_fossil[16] 0.01 0.00 0.01 -0.01 0.00 0.01 0.02 0.04 6327 1  
## x\_fossil[17] 0.02 0.00 0.00 0.02 0.02 0.02 0.03 0.03 6359 1  
## x\_fossil[18] 0.02 0.00 0.00 0.02 0.02 0.02 0.03 0.03 7163 1  
## x\_fossil[19] 0.02 0.00 0.01 0.01 0.02 0.02 0.03 0.04 5947 1  
## x\_fossil[20] 0.03 0.00 0.01 0.01 0.02 0.03 0.03 0.05 5349 1  
## x\_fossil[21] 0.02 0.00 0.00 0.01 0.02 0.02 0.02 0.03 7617 1  
## x\_fossil[22] 0.02 0.00 0.00 0.01 0.02 0.02 0.02 0.03 7566 1  
## x\_fossil[23] 0.02 0.00 0.00 0.01 0.02 0.02 0.02 0.03 8229 1  
## x\_fossil[24] 0.03 0.00 0.01 0.01 0.02 0.03 0.04 0.06 4846 1  
## y\_modern[1] 0.69 0.00 0.02 0.64 0.67 0.69 0.71 0.74 5720 1  
## y\_modern[2] 0.73 0.00 0.02 0.68 0.71 0.73 0.74 0.77 6429 1  
## y\_modern[3] 0.66 0.00 0.01 0.63 0.65 0.66 0.67 0.69 7101 1  
## y\_modern[4] 0.78 0.00 0.01 0.76 0.77 0.78 0.79 0.81 6134 1  
## y\_modern[5] 0.78 0.00 0.02 0.75 0.77 0.78 0.79 0.81 7095 1  
## y\_modern[6] 0.80 0.00 0.01 0.77 0.79 0.80 0.80 0.82 6158 1  
## y\_modern[7] 0.79 0.00 0.01 0.77 0.78 0.79 0.80 0.81 6502 1  
## y\_modern[8] 0.77 0.00 0.01 0.76 0.77 0.77 0.78 0.79 6594 1  
## y\_modern[9] 0.76 0.00 0.01 0.74 0.76 0.76 0.77 0.78 6654 1  
## y\_modern[10] 0.83 0.00 0.01 0.81 0.82 0.83 0.83 0.85 6674 1  
## y\_modern[11] 0.81 0.00 0.01 0.79 0.81 0.81 0.82 0.83 6742 1  
## y\_modern[12] 0.83 0.00 0.01 0.81 0.83 0.83 0.84 0.85 6578 1  
## y\_modern[13] 0.83 0.00 0.01 0.81 0.82 0.83 0.84 0.86 5877 1  
## y\_modern[14] 0.82 0.00 0.01 0.80 0.81 0.82 0.83 0.84 6355 1  
## y\_modern[15] 0.93 0.00 0.01 0.91 0.92 0.93 0.93 0.95 6951 1  
## y\_modern[16] 0.81 0.00 0.01 0.80 0.81 0.81 0.82 0.83 7419 1  
## y\_modern[17] 0.81 0.00 0.01 0.79 0.80 0.81 0.82 0.83 6196 1  
## y\_modern[18] 0.79 0.00 0.01 0.78 0.79 0.79 0.80 0.81 6497 1  
## y\_modern[19] 0.80 0.00 0.01 0.78 0.79 0.80 0.80 0.82 7906 1  
## y\_modern[20] 0.84 0.00 0.01 0.82 0.83 0.84 0.85 0.86 6776 1  
## y\_modern[21] 0.83 0.00 0.01 0.81 0.82 0.83 0.84 0.85 5775 1  
## y\_modern[22] 0.93 0.00 0.01 0.91 0.92 0.93 0.93 0.94 6950 1  
## y\_modern[23] 0.93 0.00 0.01 0.90 0.92 0.93 0.94 0.95 7019 1  
## y\_modern[24] 0.91 0.00 0.01 0.88 0.90 0.91 0.92 0.94 6081 1  
## y\_fossil[1] 0.16 0.00 0.01 0.14 0.15 0.16 0.16 0.17 5835 1  
## y\_fossil[2] 0.09 0.00 0.00 0.08 0.09 0.09 0.09 0.10 7499 1  
## y\_fossil[3] 0.18 0.00 0.01 0.17 0.18 0.18 0.19 0.20 6946 1  
## y\_fossil[4] 0.09 0.00 0.00 0.08 0.08 0.09 0.09 0.10 6437 1  
## y\_fossil[5] 0.09 0.00 0.00 0.08 0.09 0.09 0.09 0.10 7875 1  
## y\_fossil[6] 0.13 0.00 0.00 0.12 0.12 0.13 0.13 0.13 7564 1  
## y\_fossil[7] 0.02 0.00 0.00 0.01 0.02 0.02 0.02 0.02 9911 1  
## y\_fossil[8] 0.07 0.00 0.00 0.06 0.07 0.07 0.07 0.07 8415 1  
## y\_fossil[9] 0.02 0.00 0.00 0.01 0.02 0.02 0.02 0.03 7155 1  
## y\_fossil[10] 0.02 0.00 0.00 0.02 0.02 0.02 0.03 0.03 9085 1  
## y\_fossil[11] 0.01 0.00 0.00 0.01 0.01 0.01 0.02 0.02 7864 1  
## y\_fossil[12] 0.02 0.00 0.00 0.02 0.02 0.02 0.03 0.03 10064 1  
## y\_fossil[13] 0.02 0.00 0.00 0.01 0.01 0.02 0.02 0.02 8369 1  
## y\_fossil[14] 0.03 0.00 0.00 0.03 0.03 0.03 0.04 0.04 7697 1  
## y\_fossil[15] 0.01 0.00 0.00 0.00 0.00 0.01 0.01 0.01 7982 1  
## y\_fossil[16] 0.01 0.00 0.00 0.01 0.01 0.01 0.01 0.01 8901 1  
## y\_fossil[17] 0.02 0.00 0.00 0.01 0.01 0.02 0.02 0.02 9190 1  
## y\_fossil[18] 0.01 0.00 0.00 0.01 0.01 0.01 0.02 0.02 7373 1  
## y\_fossil[19] 0.02 0.00 0.00 0.02 0.02 0.02 0.03 0.03 9499 1  
## y\_fossil[20] 0.03 0.00 0.00 0.02 0.03 0.03 0.03 0.04 7041 1  
## y\_fossil[21] 0.01 0.00 0.00 0.00 0.01 0.01 0.01 0.01 8097 1  
## y\_fossil[22] 0.01 0.00 0.00 0.00 0.01 0.01 0.01 0.02 7280 1  
## y\_fossil[23] 0.01 0.00 0.00 0.00 0.00 0.01 0.01 0.01 8514 1  
## y\_fossil[24] 0.04 0.00 0.00 0.03 0.04 0.04 0.04 0.04 8649 1  
## int\_modern 0.99 0.00 0.00 0.99 0.99 0.99 1.00 1.00 8484 1  
## int\_fossil 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 5342 1  
## slope\_modern -3.08 0.03 0.67 -4.57 -3.51 -3.01 -2.60 -1.94 493 1  
## slope\_fossil 1.00 0.00 0.10 0.84 0.94 1.00 1.06 1.22 1733 1  
## sigma\_fossil 0.01 0.00 0.00 0.01 0.01 0.01 0.02 0.02 1420 1  
## sigma\_modern 0.07 0.00 0.02 0.03 0.05 0.07 0.08 0.12 602 1  
## inv\_m\_blank 0.25 0.00 0.04 0.18 0.22 0.25 0.28 0.34 553 1  
## m\_blank 4.11 0.03 0.68 2.96 3.62 4.04 4.53 5.63 501 1  
## F14C\_blank 0.25 0.00 0.05 0.17 0.22 0.25 0.28 0.35 620 1  
## lp\_\_ 92.94 0.42 10.15 73.61 86.14 92.75 99.73 113.04 596 1  
##   
## Samples were drawn using NUTS(diag\_e) at Fri Mar 03 13:15:19 2023.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

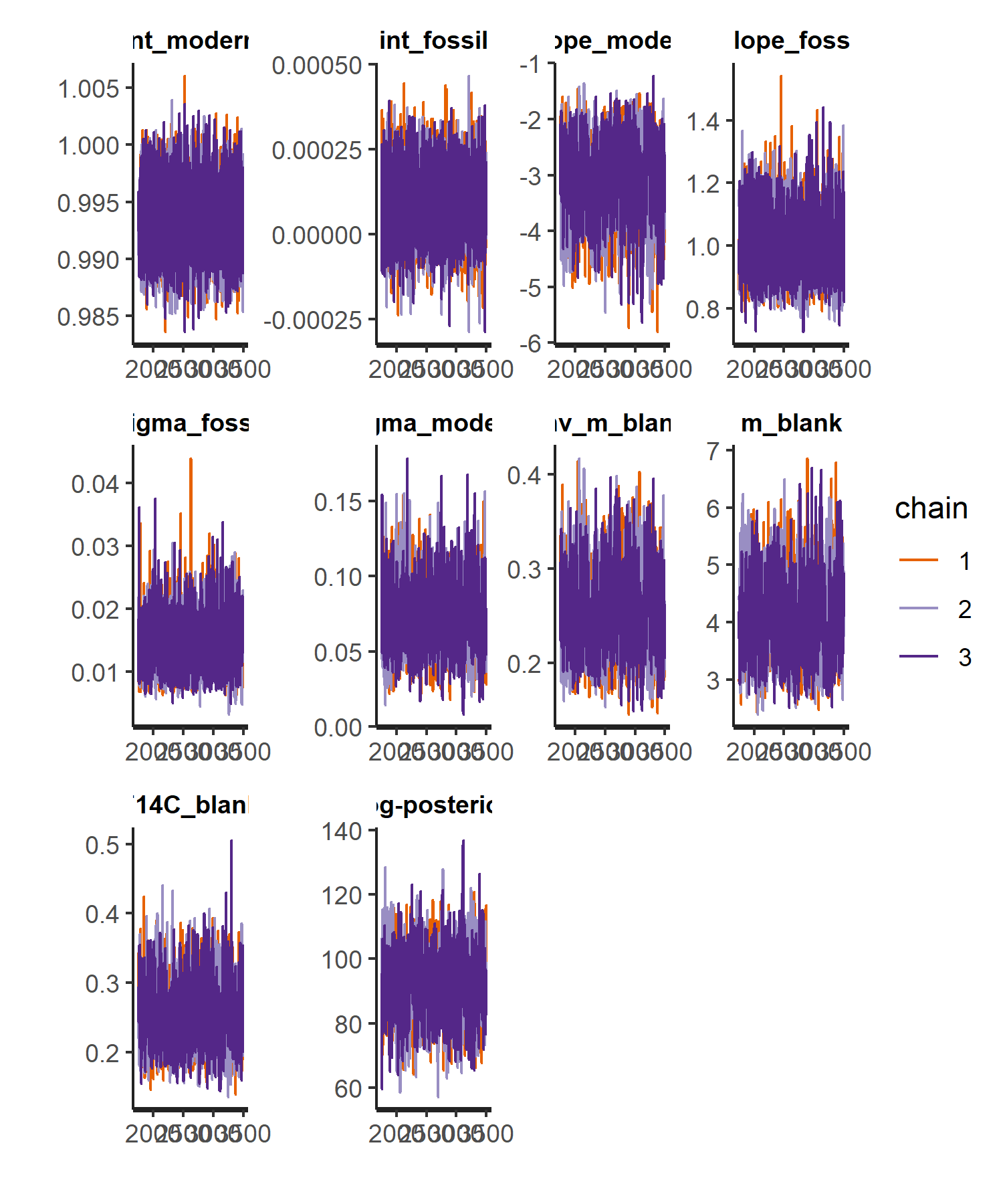
The rstan package includes default methods for plotting the fitted model, e.g., histograms of the estimated parameters and traceplots to check that the chains have converged.

rstan::stan\_hist(stan.fit.alk,  
 pars = c("x\_modern", "x\_fossil", "y\_modern", "y\_fossil"),  
 include = FALSE)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



rstan::traceplot(stan.fit.alk,  
 pars = c("x\_modern", "x\_fossil", "y\_modern", "y\_fossil"),  
 include = FALSE, inc\_warmup = FALSE)



### Extract posterior distribution

The output from a Bayesian model is the “posterior distribution” which consists of a matrix of parameters estimates. Each row gives a set of estimates from a single iteration of the sampler. There were 3 chains of 3500 iterations each, the first 2500 iterations of each chain are discarded as warm up. So we get 7500 iterations to work with.

# Extract posterior distribution  
post.dist.alk <- as\_tibble(  
 rstan::extract(stan.fit.alk,  
 pars = c("int\_modern", "int\_fossil",  
 "slope\_modern", "slope\_fossil",  
 "inv\_m\_blank", "m\_blank", "F14C\_blank")))  
  
post.dist.alk

## # A tibble: 5,250 x 7  
## int\_modern int\_fossil slope\_modern slope\_fossil inv\_m\_blank m\_blank  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.991 0.000172 -2.67 0.893 0.278 3.60  
## 2 0.993 0.000280 -3.63 0.940 0.217 4.60  
## 3 0.988 0.000144 -2.72 1.04 0.262 3.81  
## 4 0.995 0.0000607 -2.70 1.20 0.255 3.92  
## 5 0.998 0.00000421 -2.85 0.953 0.262 3.81  
## 6 0.991 -0.0000708 -3.51 1.08 0.216 4.63  
## 7 1.00 0.000164 -2.83 1.06 0.257 3.89  
## 8 0.998 0.0000608 -3.28 0.965 0.235 4.26  
## 9 0.996 0.000129 -1.96 1.11 0.324 3.08  
## 10 0.993 0.00000296 -3.17 0.984 0.239 4.18  
## # ... with 5,240 more rows, and 1 more variable: F14C\_blank <dbl>

Plot the estimated slopes as a visual check of the model. Plotting all 20000 estimates is too many so we plot a subsample.

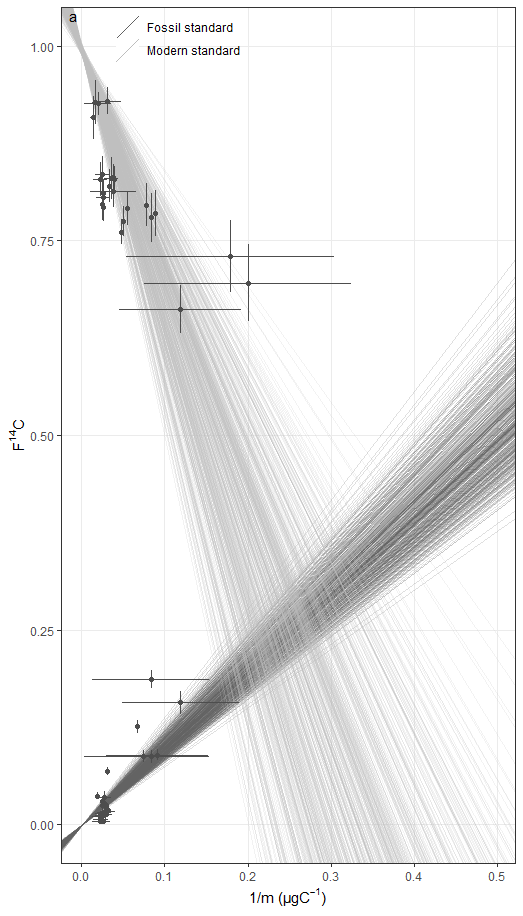
These functions recreate figures 1 from Reetz et al.

PlotPosteriorSlopes <- function(data, posterior){  
   
 # Use 2x the median value of the intersection as the axis limits  
 max.x <- 2 \* median(posterior$inv\_m\_blank)  
 max.y <- 2 \* median(posterior$F14C\_blank)  
   
 p <- ggplot(data = data,  
 aes(x = inv.mass, y = F14C, colour = sample)) +  
 geom\_abline(data = posterior,  
 aes(intercept = int\_fossil, slope = slope\_fossil,   
 colour = "Fossil standard"), alpha = 0.2) +  
 geom\_abline(data = posterior,  
 aes(intercept = int\_modern, slope = slope\_modern,  
 colour = "Modern standard"), alpha = 0.2) +  
 geom\_linerange(aes(ymax = F14C + 2\*F14C.err, ymin = F14C - 2\*F14C.err),   
 colour = "grey30", alpha = 1) +  
 geom\_segment(aes(xend = inv.mass + 2\*inv.mass.err,  
 x = inv.mass - 2\*inv.mass.err,  
 y = F14C, yend = F14C),   
 colour = "grey30", alpha = 1) +  
 geom\_point(colour = "grey30", alpha = 1) +  
 scale\_x\_continuous(name = expression("1/m (µgC"^-1\*")"),  
 limits = c(0, max.x)) +  
 scale\_y\_continuous(name = expression("F"^14\*"C"),  
 limits = c(0, 1)) +   
 scale\_colour\_manual(values = c("Fossil standard" = "#616161",  
 "Modern standard" = "#BDBDBD")) +  
 guides(color = guide\_legend(override.aes = list(alpha = 5))) +   
 theme\_bw() +   
 theme(legend.position = c(0, 1),   
 legend.justification = c(-0.35, 0.9),   
 legend.title = element\_blank(),  
 legend.background = element\_blank(),  
 panel.grid.minor = element\_blank())  
 return(p)  
}  
  
PlotIntersections <- function(posterior){  
 upr <- function(x){median(x) + mad(x)}  
 lwr <- function(x){median(x) - mad(x)}  
   
 # Use 2x the median value of the intersection as the axis limits  
 max.x <- 2 \* median(1/posterior$inv\_m\_blank)  
 max.y <- 2 \* median(posterior$F14C\_blank)  
   
 p.w <- dplyr::summarise\_all(posterior,  
 funs(median = median, sd = sd, mad = mad,  
 lwr = lwr, upr = upr))   
 p.ints <- ggplot(data = posterior) +  
 geom\_point(aes(x = m\_blank, y = F14C\_blank), alpha = 0.02, size = 1.5) +  
 geom\_linerange(data = p.w,  
 aes(x = m\_blank\_median, y = F14C\_blank\_median,  
 ymax = F14C\_blank\_upr, ymin = F14C\_blank\_lwr),  
 colour = "gray0", size = 1, alpha = 0.8) +  
 geom\_errorbarh(data = p.w,  
 aes(x = m\_blank\_median, y = F14C\_blank\_median,  
 xmax = m\_blank\_upr, xmin = m\_blank\_lwr,  
 height = 0), colour = "gray0", size = 1, alpha = 0.8) +  
 scale\_x\_continuous(expression("m"[blank]\*" (µgC)"), limits = c(0, max.x)) +  
 scale\_y\_continuous(expression("F"^14\*"C"), limits = c(0, max.y)) +  
 theme\_bw() +  
 theme(panel.grid.minor = element\_blank())  
 p.ints  
}

# Number of lines to plot  
n.samples <- 500  
  
post.dist.samp.alk <- post.dist.alk[sample(1:nrow(post.dist.alk), n.samples), ]  
  
p.slps.alk <- PlotPosteriorSlopes(data = alkane,  
 posterior = post.dist.samp.alk) +  
 annotate("text", label = "a", x = -Inf, y = Inf, hjust = -1, vjust = 1.2)  
p.slps.alk

## Warning: Removed 2 rows containing missing values (geom\_segment).

## Warning: Removed 21 rows containing missing values (geom\_segment).



Plot all the intersections of the two regression lines. The intersections are the paired estimates of F14C\_blank and inv\_m\_blank.

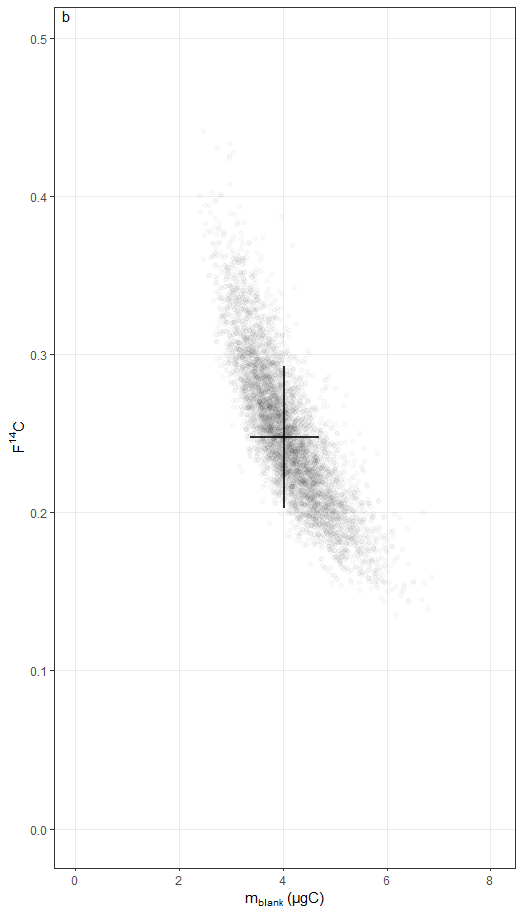
p.ints.alk <- PlotIntersections(post.dist.alk) +  
 annotate("text", label = "b", x = -Inf, y = Inf, hjust = -1, vjust = 1.2)

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was generated.

## Warning: Ignoring unknown aesthetics: x

p.ints.alk

## Warning: Removed 1 rows containing missing values (geom\_point).



A combined figure.

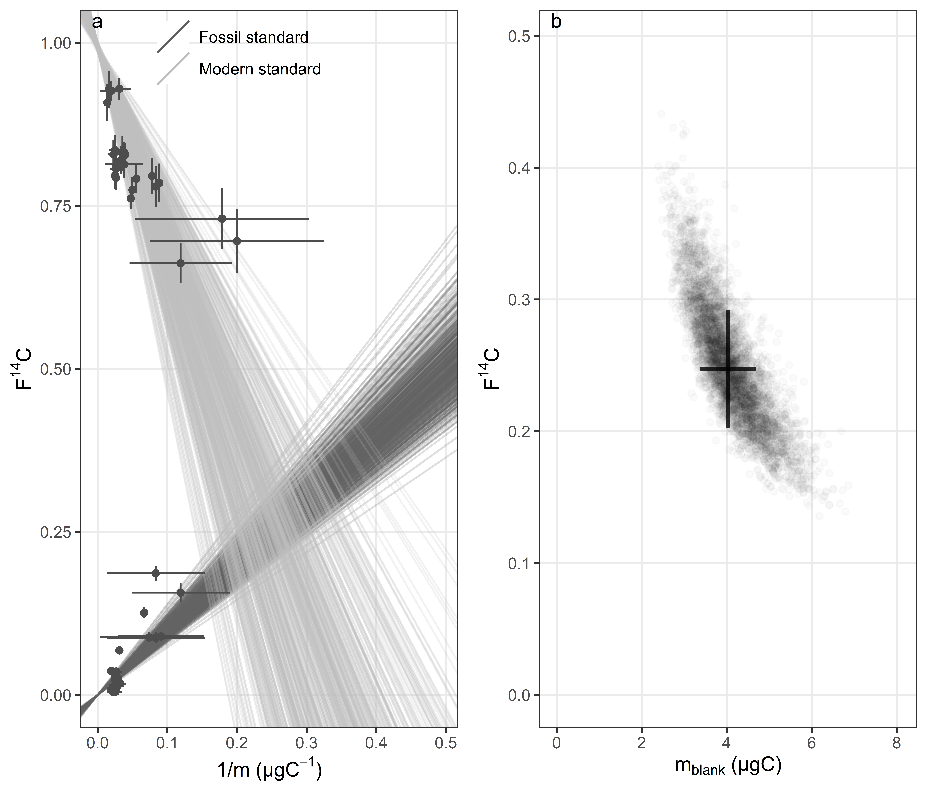
p.slps.ints.alk <- egg::ggarrange(p.slps.alk, p.ints.alk, ncol = 2, draw = FALSE)

## Warning: Removed 2 rows containing missing values (geom\_segment).

## Warning: Removed 21 rows containing missing values (geom\_segment).

## Warning: Removed 1 rows containing missing values (geom\_point).

p.slps.ints.alk



ggsave("p.slps.ints.alk.tiff", p.slps.ints.alk, height = 3, width = 6.5)  
ggsave("p.slps.ints.alk.png", p.slps.ints.alk, height = 3, width = 6.5)

####ggsave(“p.slps.ints.alk.png”, p.slps.ints.alk, height = 6.0, width = 7.0, units = c(“in”), dpi = 1500)

## Summary statistics

The median, 2.5% and 97.5% quantiles can be used as point estimates + CI of the values of inv\_m\_blank and F14C\_blank. The median absolute deviation (MAD) might be better than the standard deviation as a measure of uncertainty for error propagation because the intersection is the ratio of the differences in slopes and intercepts, so its distribution has very long tails (is not Gaussian).

tab.alk <- post.dist.alk %>%  
 gather(parameter, value) %>%  
 group\_by(parameter) %>%  
 summarise\_all(funs(mean = mean, sd = sd, median = median, MAD = mad,  
 lwr = quantile(., 0.025), upr = quantile(., 0.975))) %>%   
 mutate("Standard compound" = "n-alkane") %>%   
 select("Standard compound", everything())  
  
knitr::kable(tab.alk, digits = 3)

| Standard compound | parameter | mean | sd | median | MAD | lwr | upr |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n-alkane | F14C\_blank | 0.251 | 0.045 | 0.247 | 0.045 | 0.172 | 0.350 |
| n-alkane | int\_fossil | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| n-alkane | int\_modern | 0.994 | 0.003 | 0.994 | 0.003 | 0.988 | 1.000 |
| n-alkane | inv\_m\_blank | 0.250 | 0.041 | 0.248 | 0.041 | 0.178 | 0.338 |
| n-alkane | m\_blank | 4.107 | 0.680 | 4.036 | 0.662 | 2.957 | 5.629 |
| n-alkane | slope\_fossil | 1.004 | 0.098 | 0.996 | 0.094 | 0.839 | 1.219 |
| n-alkane | slope\_modern | -3.077 | 0.672 | -3.006 | 0.660 | -4.566 | -1.942 |