Supporting Information for:

Evaluation of Air-Free Glassware Using the Ketyl Test

Alexandra W. Carlson, Danielle A. Primka, Ellis D. Douma, Miriam A. Bowring*

Table of Contents

1. Photos of experimental apparatus

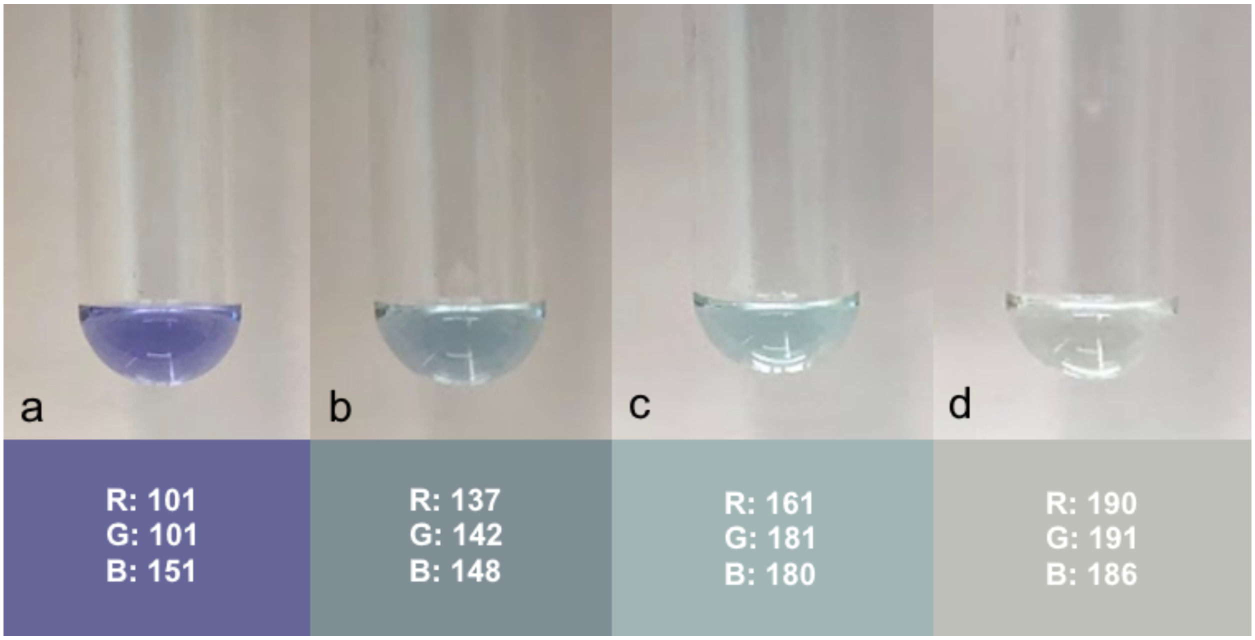

Figure S1. Sample image frames used for ketyl test color analysis of glassware **A**, with RGB values given below on a background of the reported color. Frame **a** was captured at 12 minutes after exposure to ambient atmosphere with RGB values of 101, 101, and 151. Frame **b** was captured at 25 minutes with RGB values of 137, 142, and 148. Frame **c** was captured at 165 minutes with RGB values of 161, 181, and 180. Frame **d** was captured at 301 minutes with RGB values of 190, 101, and 186.

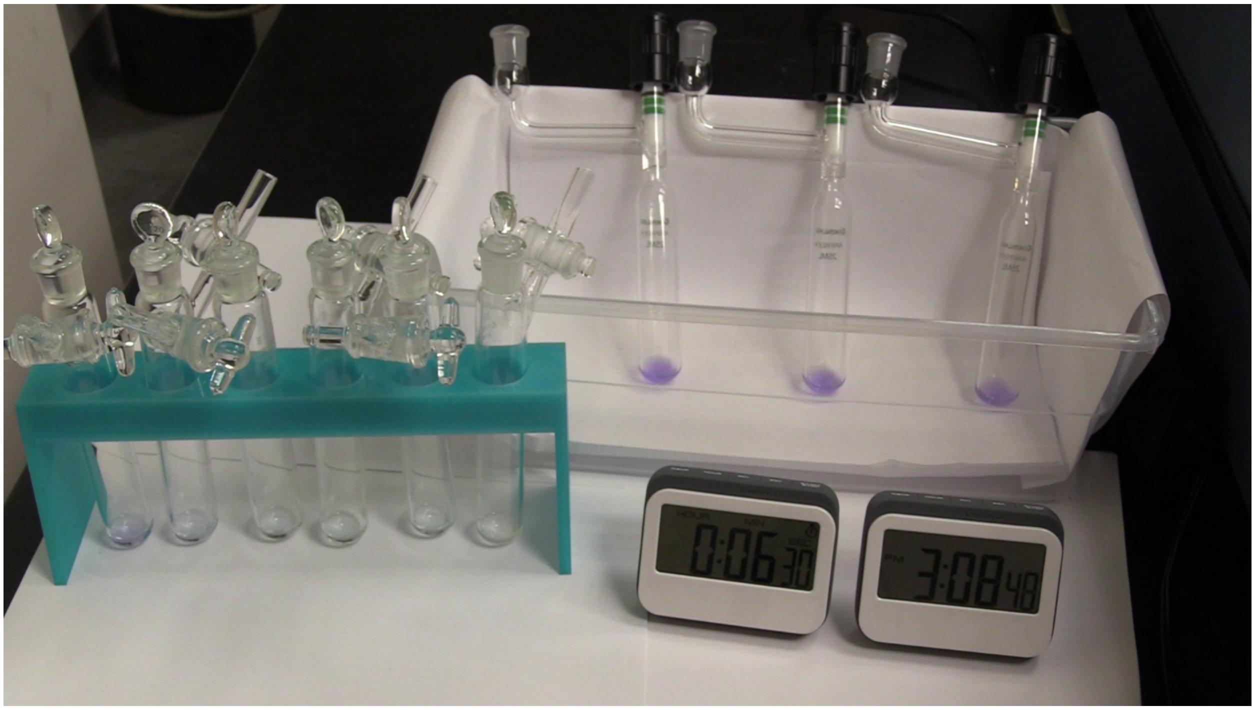

Figure S2. Sample experimental setup for glassware **A** (left) and **D** (right).

2. Data for additional samples

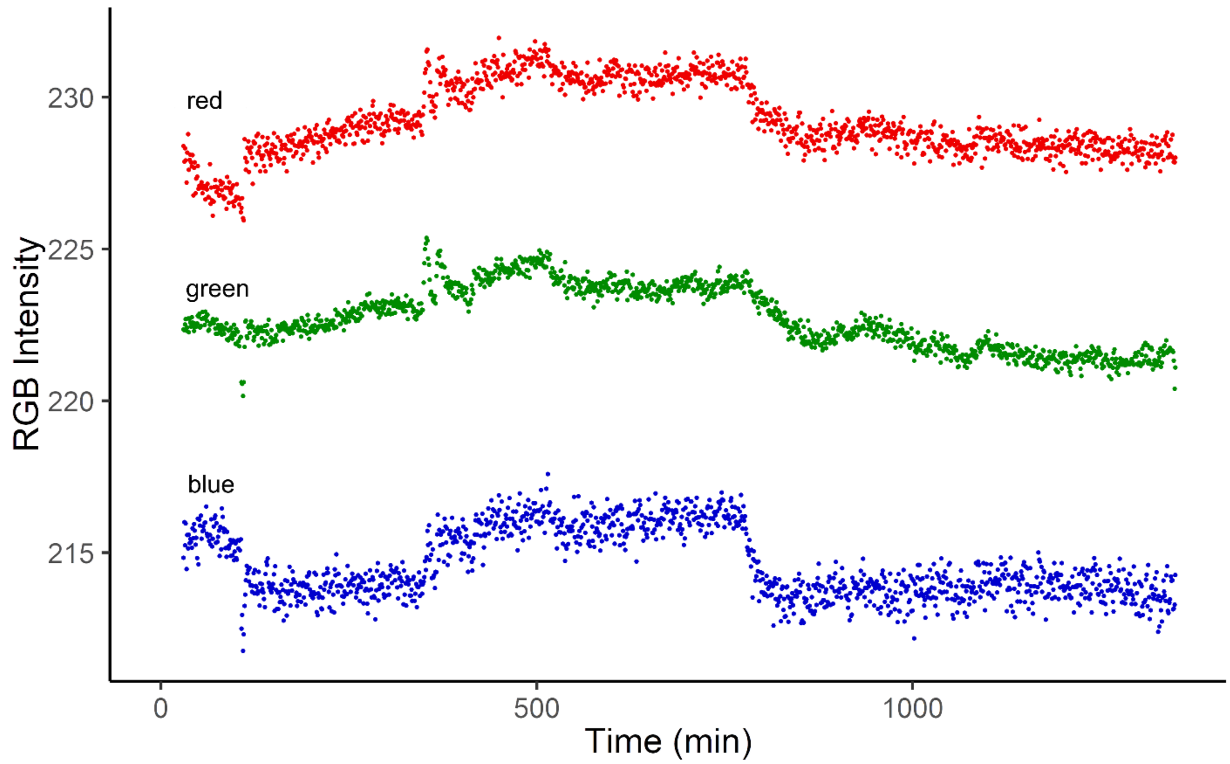

Figure S3. Red, green, and blue color values from video of the ketyl test solution in a 25 mL Schlenk tube with no stopper.

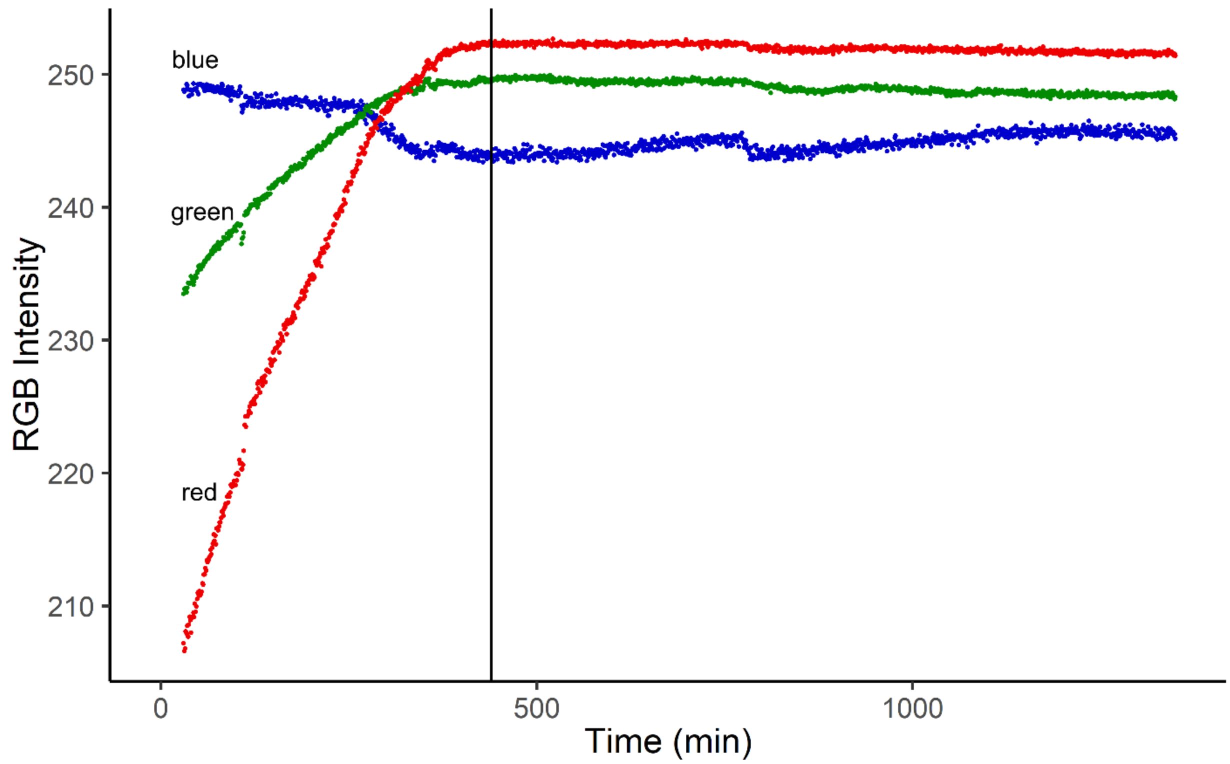

Figure S4. Red, green, and blue color values from video of the ketyl test solution in a 50 mL Schlenk tube with greased ground glass stopcock and stopper (**B**). The vertical line indicates the point at which the solution is defined as colorless based on change point analysis.

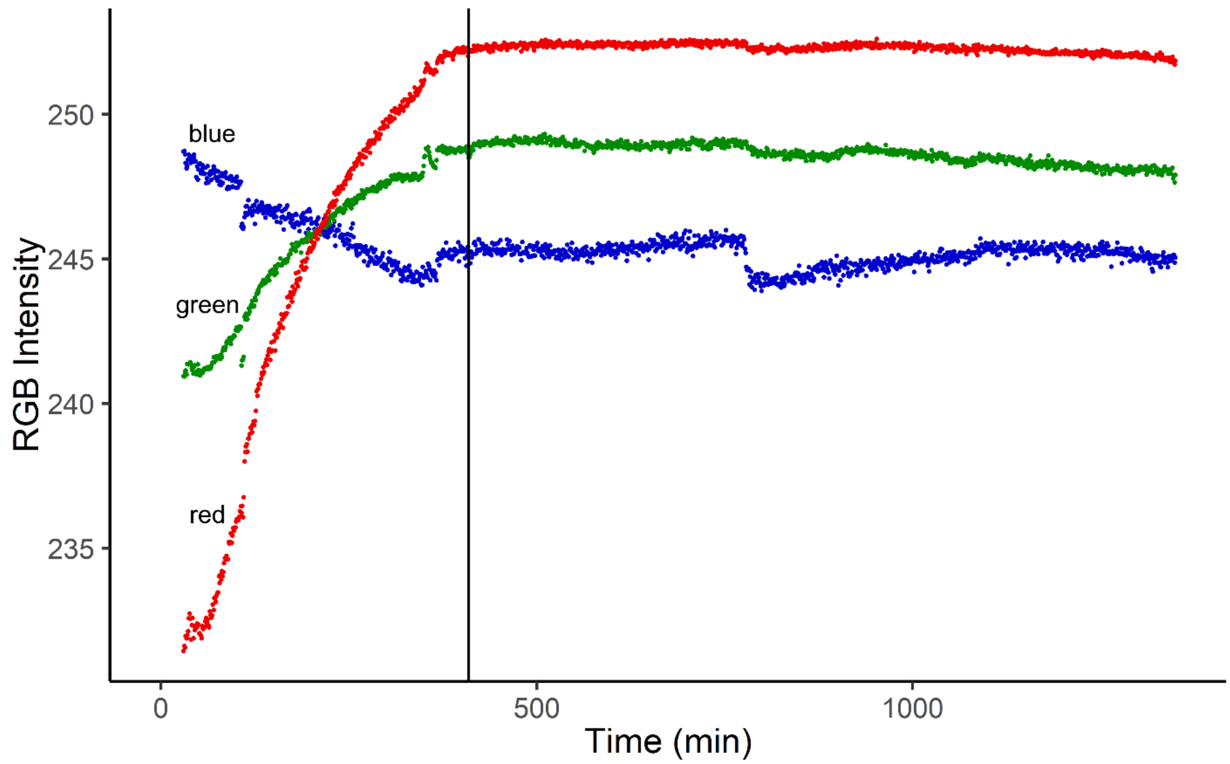

Figure S5. Red, green, and blue color values from video of the ketyl test solution in a 50 mL round-bottom Schlenk flask with greased ground glass stopcock and stopper (**C**). The vertical line indicates the point at which the solution is defined as colorless based on change point analysis.

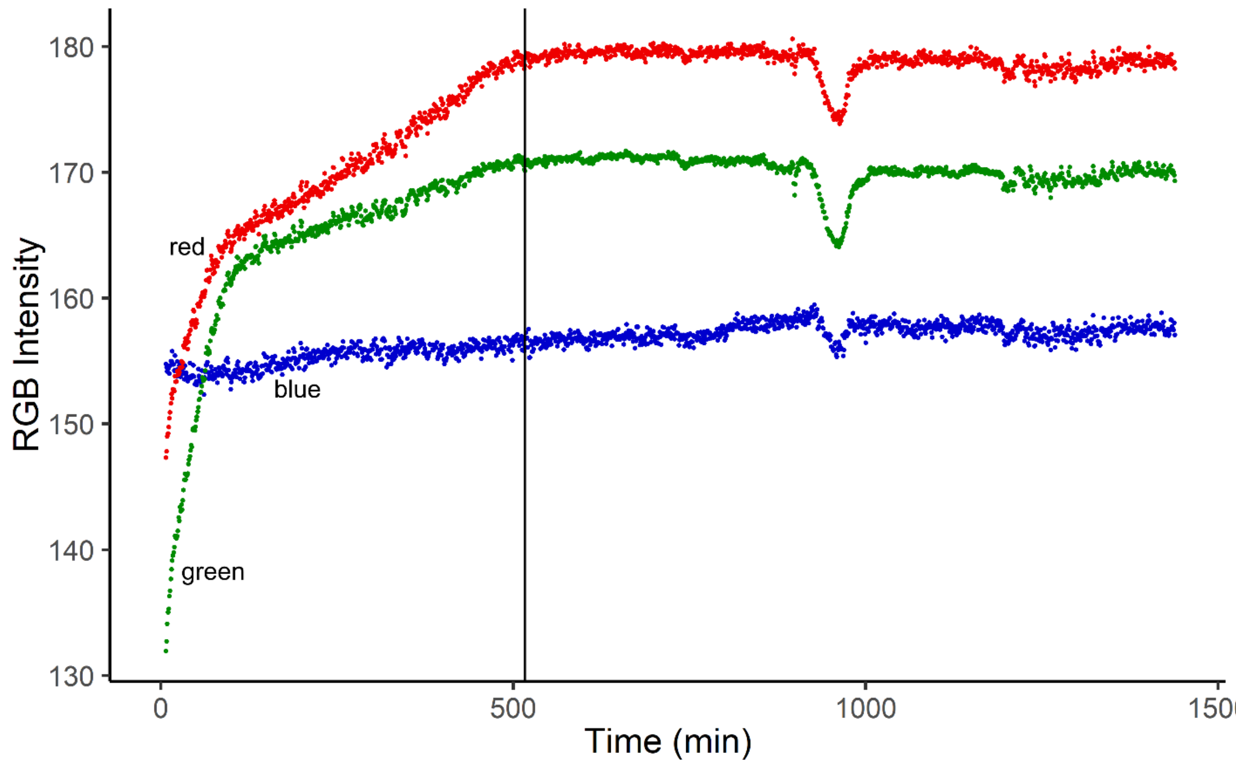

Figure S6. Red, green, and blue color values from video of the ketyl test solution in a 25 mL Schlenk tube with threaded PTFE stopper (**D**). The vertical line indicates the point at which the solution is defined as colorless based on change point analysis.

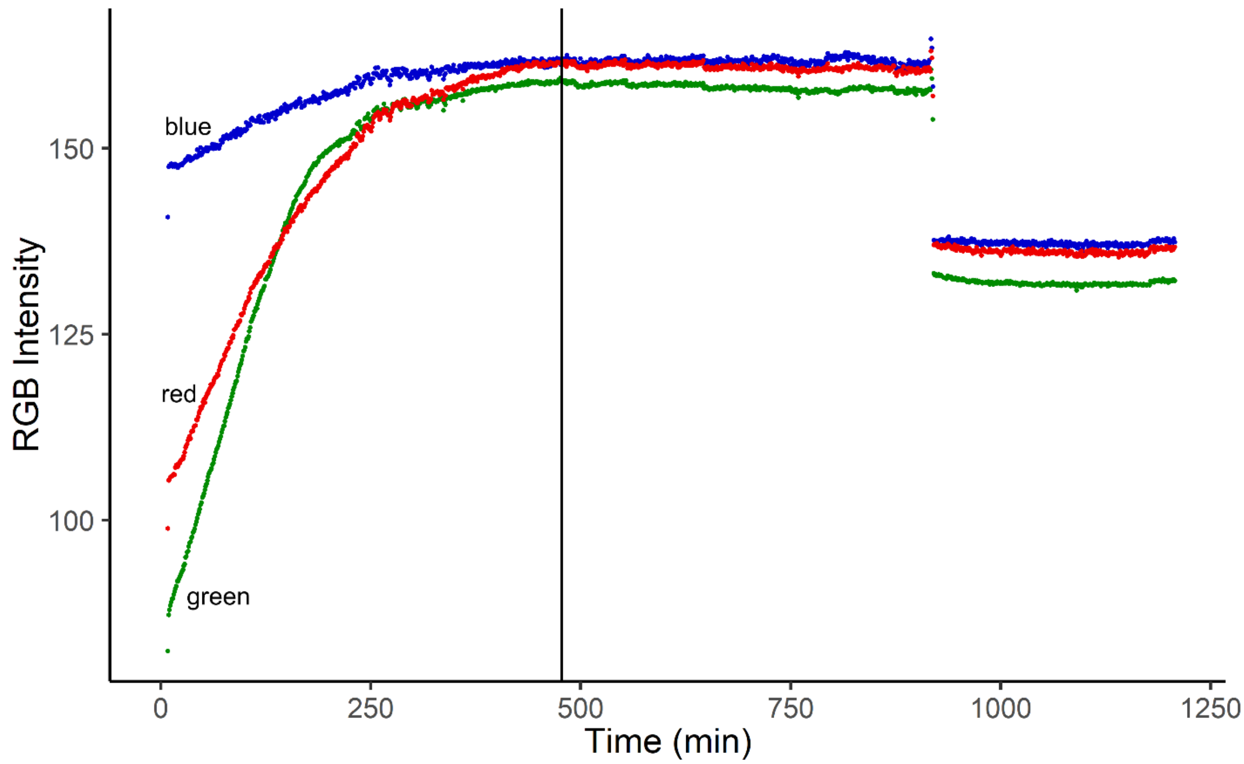

Figure S7. Red, green, and blue color values from video of the ketyl test solution in an NMR tube with standard cap (**E**). The vertical line indicates the point at which the solution is defined as colorless based on change point analysis.

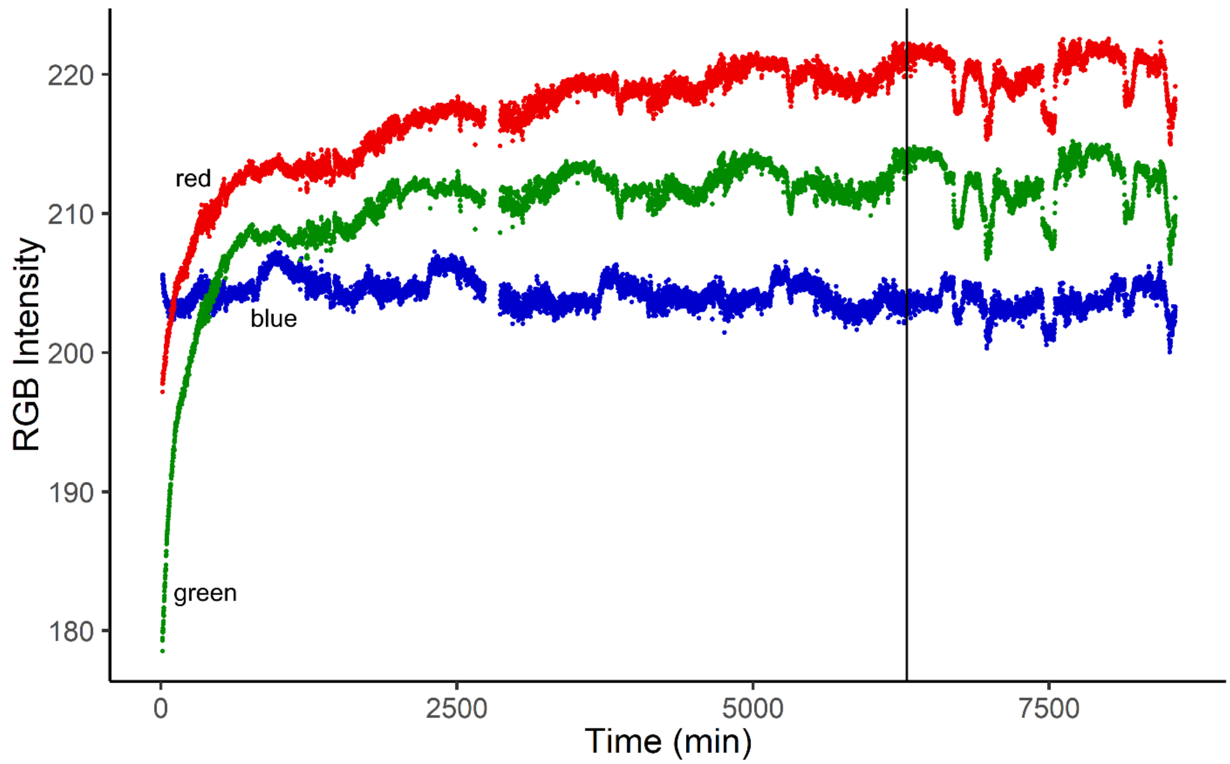

Figure S8. Red, green, and blue color values from video of the ketyl test solution in a LPV NMR tube with threaded PTFE stopper (**F**). The vertical line indicates the point at which the solution is defined as colorless based on visual estimation.

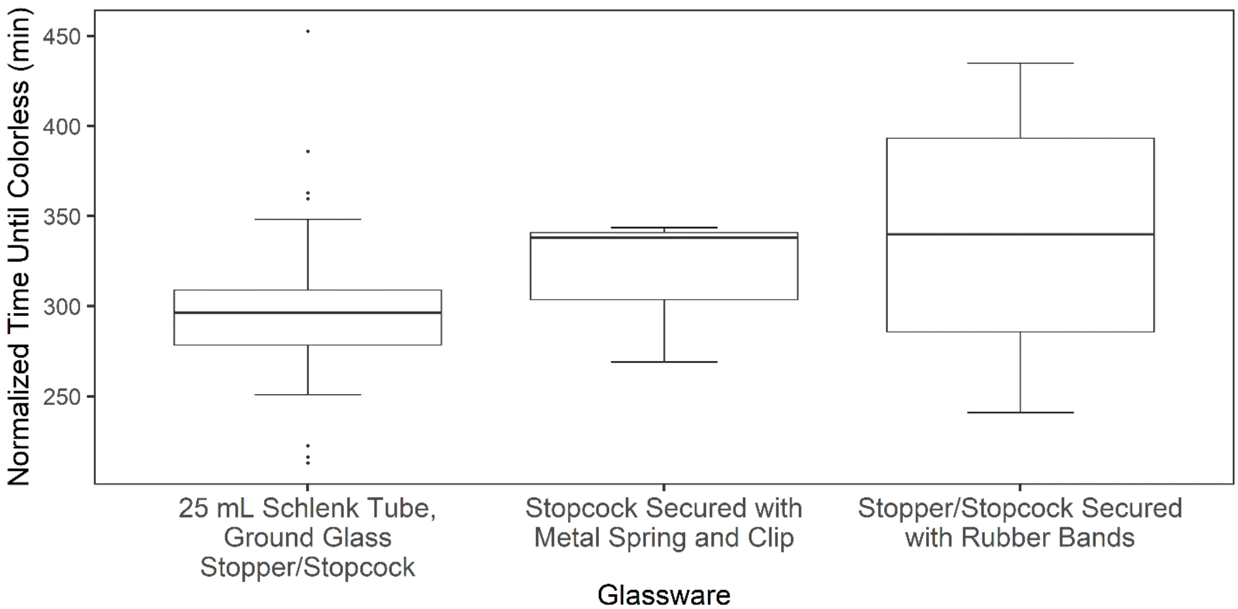

Figure S9. Box plot of times until colorless for 25 mL Schlenk tubes with unsecured ground glass stoppers/stopcocks **A**, stopcocks secured with a metal spring and clip, and stoppers/stopcocks secured with rubber bands. Times were normalized to 300 min for concurrently monitored standard 25 mL Schlenk tubes with ground glass stoppers.

Table S1. Normalized mean time until colorless for different methods of securing stopcocks. The unsecured stopcocks were tested over 65 trials, and six trials were performed for each stopcock securing method. Times were normalized to 300 min for concurrently monitored standard 25 mL Schlenk tubes with ground glass stopcocks and stoppers.

Stopper/Stopcock	Time Until Colorless (min)
Unsecured	300 ± 10
Stopcock secured with metal spring and clip	317 ± 103
Stopper and stopcock secured with rubber bands	339 ± 80

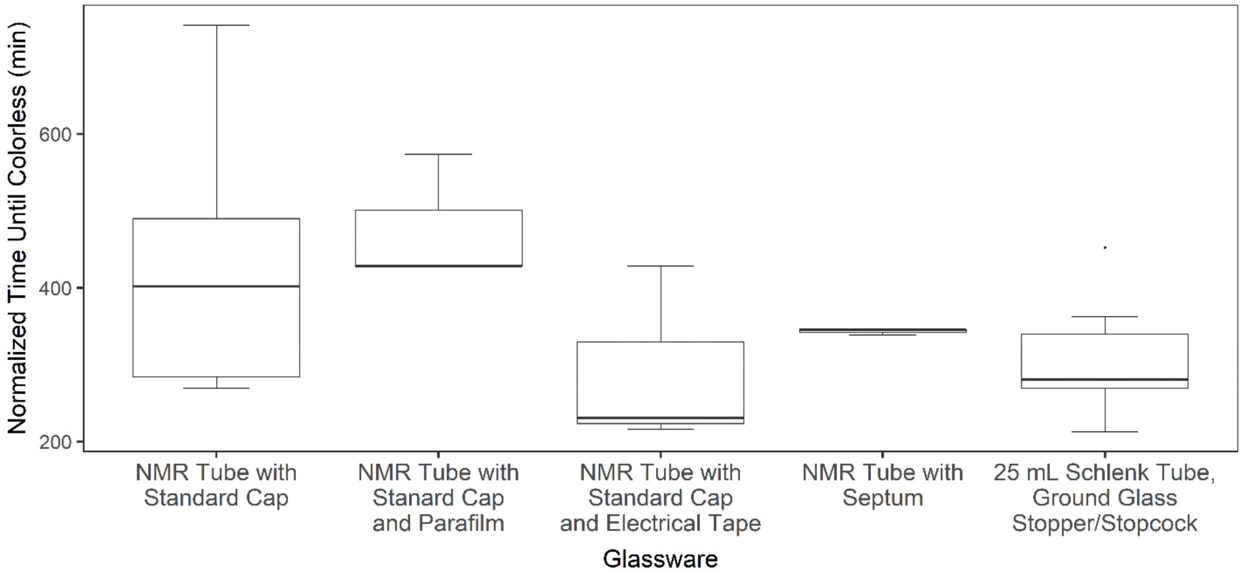

Figure S10. Box plot of times until colorless for NMR tubes with standard caps **E**, NMR tubes with standard caps and parafilm, NMR tubes with standard caps and electrical tape, NMR tubes with septa, and 25 mL Schlenk tubes with ground glass stoppers **A**. Times were normalized to 300 min for concurrently monitored standard 25 mL Schlenk tubes with ground glass stoppers.

Table S2. Normalized mean time until colorless for different methods of capping NMR tubes. NMR tubes with standard caps were tested over 10 trials, and three trials were performed for each alternative capping method. Times were normalized to 300 min for concurrently monitored standard 25 mL Schlenk tubes with ground glass stopcocks and stoppers.

NMR Tube Cap	Time Until Colorless (min)
Standard cap	413 ± 106
Standard cap and parafilm	477 ± 209
Standard cap and electrical tape	292 ± 294
Septum	343 ± 10

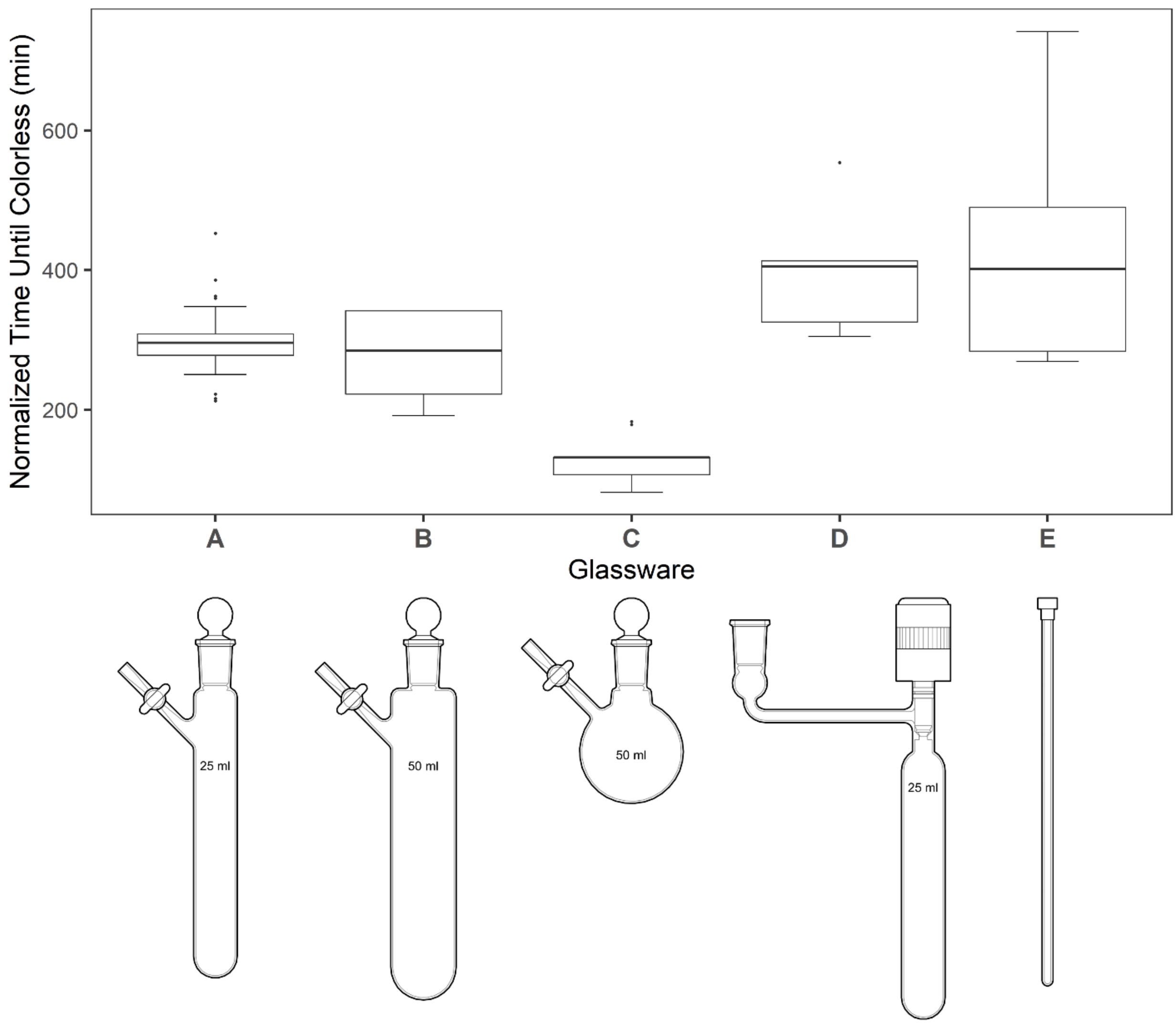

Figure S11. Box plot of ketyl test times until colorless for different shapes and sizes of Schlenk glassware. **A**, 25 mL Schlenk tube, ground glass stopper. **B**, 50 mL Schlenk tube, ground glass stopper. **C**, 50 mL round-bottom Schlenk flask, ground glass stopper**. D**, 25 mL Schlenk tube, threaded PTFE stopper. **E**, NMR tube with standard cap. Times were normalized to 300 min for concurrently monitored standard 25 mL Schlenk tubes with greased ground glass stoppers **A**. These data are the same as plotted in Figure 2 of the main text, but with a narrower range on the vertical axis to show detail.

3. ImageJ Macros

```
macro "Stack RGB Measurer" {
if (nSlices==1)
exit("Stack required");
stack1 = getImageID;
n = nSlices;
red = newArray(0);
green = newArray(0);
blue = newArray(0);
for (i=1; i<=n; i++) {
showProgress(i, n);
selectImage(stack1);
setSlice(i);
setRGBWeights(1, 0, 0);
red = Array.concat(red, getValue("Mean"));
setRGBWeights(0, 1, 0);
green = Array.concat(green, getValue("Mean"));
setRGBWeights(0, 0, 1);
                    blue = Array.concat(blue, getValue("Mean"));
}
Array.show("Results", red, green, blue);
setOption("ShowRowNumbers", true);
updateResults;
setSlice(1);
}
```

```
macro "Stack RGB Measurer All ROIs" {
if (nSlices==1)
exit("Stack required");
stack1 = getImageID;
n = nSlices;
m = roiManager("count");
for (j=0; j<m; j++) {
roiManager("select", j);
red = newArray(0);
green = newArray(0);
blue = newArray(0);
for (i=1; i<=n; i++) {
showProgress(i, n);
selectImage(stack1);
setSlice(i);
setRGBWeights(1, 0, 0);
red = Array.concat(red, getValue("Mean"));
setRGBWeights(0, 1, 0);
green = Array.concat(green, getValue("Mean"));
setRGBWeights(0, 0, 1);
blue = Array.concat(blue, getValue("Mean"));
}
Array.show(Roi.getName, red, green, blue);
Table.showRowNumbers(true, Roi.getName);
Table.update(Roi.getName);
setSlice(1);
}
}
```

4. R Functions

```
### load packages (install first via right hand side if not installed)
library(knitr, pandoc)
library(tidyverse)
library(plotly)
```

basic functions

```
## input two datasets and the time gap in minutes between the last image of the first one and the first image of the second one
## returns a joined/combined dataset
join <- function(data1, data2, gap) {
sgap <- gap * 0.999 # had to convert the minutes into "snapshots"
n <- max(data1$X) + sgap
data2$X <- data2$X + n
newdata <- bind_rows(data1, data2)
return(newdata)
}

## input entire dataset and the number of minutes at first image
## returns a dataset converted to minutes
convert <- function(data, minutes) {
data$X <- ((data$X - 1) * (30/29.97)) + minutes # conversion comes from selection of image sequences and frame rate
return(data)
}
```

findpoint functions

```
## input entire dataset and the number of minutes at first image
## returns a table of means
means <- function(data, minutes) {
data <- convert(data, minutes)
n <- nrow(data)
i <- 1
red.means <- read.csv(text = "start, mean")
while (i <= (n - 29)) { # this is determined by the sizes of the chunks
x <- mean(data$red[i:(i+29)]) # this sets the size of the chunks
red.means <- add_row(red.means, start = data$X[i], mean = x)
i <- i + 10 # this sets the size of the offset
}
return(red.means)
}

## input entire data and the number of minutes at first image
## returns a table of differences of means
diffs <- function(data, minutes) {
meantable <- means(data, minutes)
n <- nrow(meantable)
i <- 1
red.diffs <- read.csv(text = "start, difference")
while (i < n) {
diff <- meantable$mean[(i+1)] - meantable$mean[i]
red.diffs <- add_row(red.diffs, start = meantable$start[i], difference = diff)
i <- i + 1
}
return(red.diffs)
}

## input entire dataset and the number of minutes at first image
## finds the first value under a set number
## returns the middle point of the found difference of means
findpoint <- function(data, minutes) {
difftable <- diffs(data, minutes)
n <- nrow(difftable)
i <- 1
while (i < n) {
```

```
if (difftable$difference[i] <= 0.05) { # this is the limit for the difference
point <- difftable$start[i] + (19 * (30/29.97)) # this gives the middle of the two chunks,
accounting for the conversion into minutes
return(point)
break
}
i <- i + 1
}
return(0) # if the function fails to find a difference that meets the requirement it'll return 0
}

## input a named list of datasets and the number of minutes at first image
## returns a table of times until colorless in minutes
times <- function(datalist, minutes) {
i <- 1
l <- length(datalist)
timetable <- read.csv(text = "sample, time")
while (i <= l) {
point <- findpoint(datalist[[i]], minutes)
timepoint <- round(point)
timetable <- add_row(timetable, sample = names(datalist[i]), time = timepoint)
i <- i + 1
}
return(timetable)
}
```

plot functions

```
# to make a plot interactive, use ggplotly(plotname)

## input entire dataset and the number of minutes at first image
## returns a plot of RGB data without the line
allplot <- function(data, minutes) {
data <- convert(data, minutes)
m <- max(data$X)
plot <- ggplot(data, aes(x = X, y = blue)) + geom_point(color = "blue3", size = 1) +
geom_point(aes(x = X, y = green), color = "green4", size = 1) + geom_point(aes(x = X, y = red),
color = "red2", size = 1) + labs(x = "Time (min)", y = "RGB Intensity") + theme_classic() +
coord_cartesian(xlim = c(0, m))
return(plot)
}

## input entire dataset and the number of minutes at first image
## returns a plot of RGB data with the line marked
allpointplot <- function(data, minutes) {
point <- findpoint(data, minutes)
data <- convert(data, minutes)
m <- max(data$X)
plot <- ggplot(data, aes(x = X, y = blue)) + geom_point(color = "blue3", size = 1) +
geom_point(aes(x = X, y = green), color = "green4", size = 1) + geom_point(aes(x = X, y = red),
color = "red2", size = 1) +  geom_vline(xintercept = point) + labs(x = "Time (min)", y = "RGB
Intensity") + theme_classic() + coord_cartesian(xlim = c(0, m))
return(plot)
}

## input entire dataset and the number of minutes at first image
## returns a plot of the red data with the line marked
redplot <- function(data, minutes) {
point <- findpoint(data, minutes)
data <- convert(data, minutes)
m <- max(data$X)
plot <- ggplot(data, aes(x = X, y = red)) + geom_point(color = "red2", size = 1) +
geom_vline(xintercept = point) + labs(x = "Time (min)", y = "Red Intensity") + theme_classic()
+ coord_cartesian(xlim = c(0, m))
return(plot)
}
```

```
## input entire dataset and the number of minutes at first image
## returns a plot of the red data with the line marked, with the x-axis starting where the data
starts
altredplot <- function(data, minutes) {
point <- findpoint(data, minutes)
data <- convert(data, minutes)
l <- min(data$X)
m <- max(data$X)
plot <- ggplot(data, aes(x = X, y = red)) + geom_point(color = "red2", size = 1) +
geom_vline(xintercept = point) + labs(x = "Time (min)", y = "Red Intensity") + theme_classic()
+ coord_cartesian(xlim = c(l, m))
return(plot)
}

## input a named list of data sets and the number of minutes at first image
## returns a list of red plots
redplotlist <- function(datalist, minutes) {
i <- 1
l <- length(datalist)
plotlist <- list()
names <- names(datalist)
while (i <= l) {
plotlist[[i]] <- redplot(datalist[[i]], minutes) + labs(title = names[i])
i <- i + 1
}
return(plotlist)
}
```