Persistence Experiments

Christian Cole

10/03/2020

Analysis of Persistence Experiments

Go through experiment sets and extract the persistence data. The data are defined by having "ObsType" != "Ndata".

```
# Read in ML's data
my.dat = read_xlsx("dat/20191030DatasetPaper.xlsx", sheet = "DataMl")

# only use the persistence experiments for Groups 1 & 2. Groups 3-6 do not
# have enough replicates to be meaningful
persist.dat = my.dat %>% filter(
      Substrate != 'Cott01' &
      Group %in% c(1,2) &
      ObsType != 'Ndata') %>%
  select(c("Substrate", "ObsType", "Count", "Mass", "Duration", "Time", "Group", "Repeat")) %>%
  rename(ObservationType = ObsType,
         TransferTime = Duration,
         PersistenceTime = Time,
         Experiment = Group,
         Replicate = Repeat)

# fix column types
persist.dat$Count = as.numeric(persist.dat$Count)
persist.dat$PersistenceTime = as.numeric(persist.dat$PersistenceTime)
persist.dat$Experiment = as.character(persist.dat$Experiment)
persist.dat$Replicate = as.character(persist.dat$Replicate)
# some errors in the data, fix
persist.dat[persist.dat$Substrate == 'wool01', 'Substrate'] <- 'Wool'
persist.dat[persist.dat$Substrate == 'Wool01', 'Substrate'] <- 'Wool'
persist.dat[persist.dat$Substrate == 'Nylo01', 'Substrate'] <- 'Nylon'
persist.dat[persist.dat$Substrate == 'Wool' & persist.dat$Experiment == '2', 'Substrate'] <- 'Nylon'

# summarise the count data
summ.dat = summarySE(persist.dat, measurevar = "Count", groupvars = c("Substrate", "PersistenceTime", "E
knitr::kable(head(summ.dat), caption = "Summary of ML's Persistence Data")
```

Table 1: Summary of ML's Persistence Data

Substrate	PersistenceTime	Experiment	N	Count	sd	se	ci
Nylon	0	2	6	49.83333	24.449267	9.981371	25.657932
Nylon	30	2	6	30.33333	22.033308	8.995060	23.122539
Nylon	60	2	6	22.00000	14.615061	5.966574	15.337566

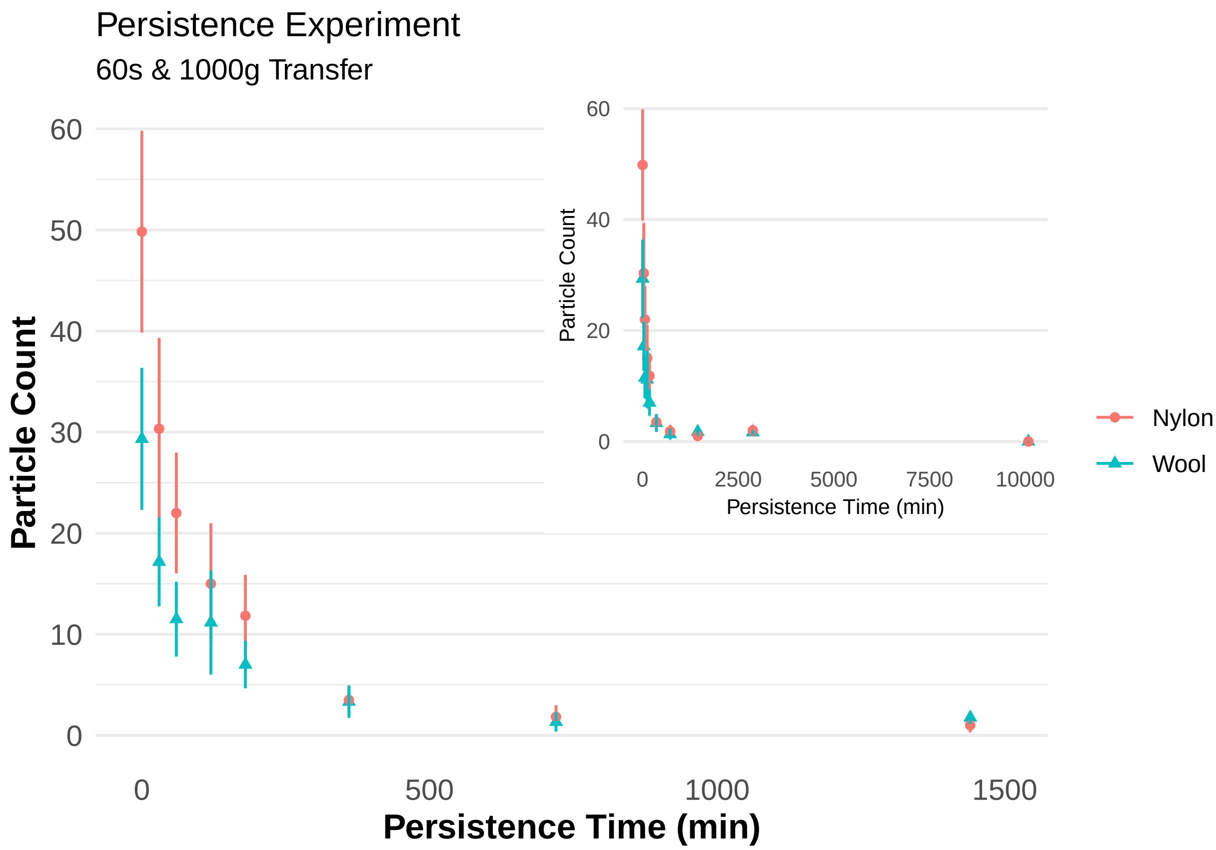
Substrate	PersistenceTime	Experiment	N	Count	sd	se	ci
Nylon	120	2	6	15.00000	14.669697	5.988879	15.394903
Nylon	180	2	6	11.83333	9.907909	4.044887	10.397713
Nylon	360	2	6	3.50000	3.507136	1.431782	3.680513

The data covers a large period of time - potentially weeks - so the time scale needs to be compressed somehow for ease of interpretation. Let's see what we can do.

```
p1 = ggplot(summ.dat, aes(x=PersistenceTime, y = Count, group = Substrate, colour = Substrate, shape=Sul
  geom_point(position = position_dodge(0.3)) +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08) +
  xlim(c(-5,1500)) +
  scale_y_continuous(breaks = seq(0,60,10)) +
  labs(title = "Persistence Experiment",
       subtitle = "60s & 1000g Transfer",
       x = "Persistence Time (min)",
       y = "Particle Count") +
  mytheme

p2 = ggplot(summ.dat, aes(x=PersistenceTime, y = Count, group = Substrate, colour = Substrate, shape=Sul
  geom_point(position = position_dodge(0.3)) +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08) +
  labs(x = "Persistence Time (min)",
       y = "Particle Count") +
  inset_theme

p1 + annotation_custom(ggplotGrob(p2), xmin = 700, xmax = 1600,
                       ymin = 20, ymax = 65)
```



The data seems to be a smooth decay. Let's try some curve-fitting.

This function SSasymp() a self-starting fucntion which attempts to find suitable initial parameters for perform a fit - here a non-linear least squares.

Let's see the Nylon fit first

```r
# ideas taken from http://douglas-watson.github.io/post/2018-09_exponential_curve_fitting/
# and https://dataconomy.com/2017/08/nonlinear-least-square-nonlinear-regression-r/
set.seed(12345)
ny.fit = nls(Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha),
             data = summ.dat,
             subset = Substrate == 'Nylon')
ny.fit.err = summ.dat$Count-predict(ny.fit)
ny.nlm_error <- sqrt(mean(ny.fit.err^2))
ny.fit
```

```
## Nonlinear regression model
##   model: Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha)
##    data: summ.dat
##    Countf    Count0 log_alpha
##     2.060    47.063    -4.458
##  residual sum-of-squares: 52.85
##
## Number of iterations to convergence: 0
## Achieved convergence tolerance: 5.673e-06
```

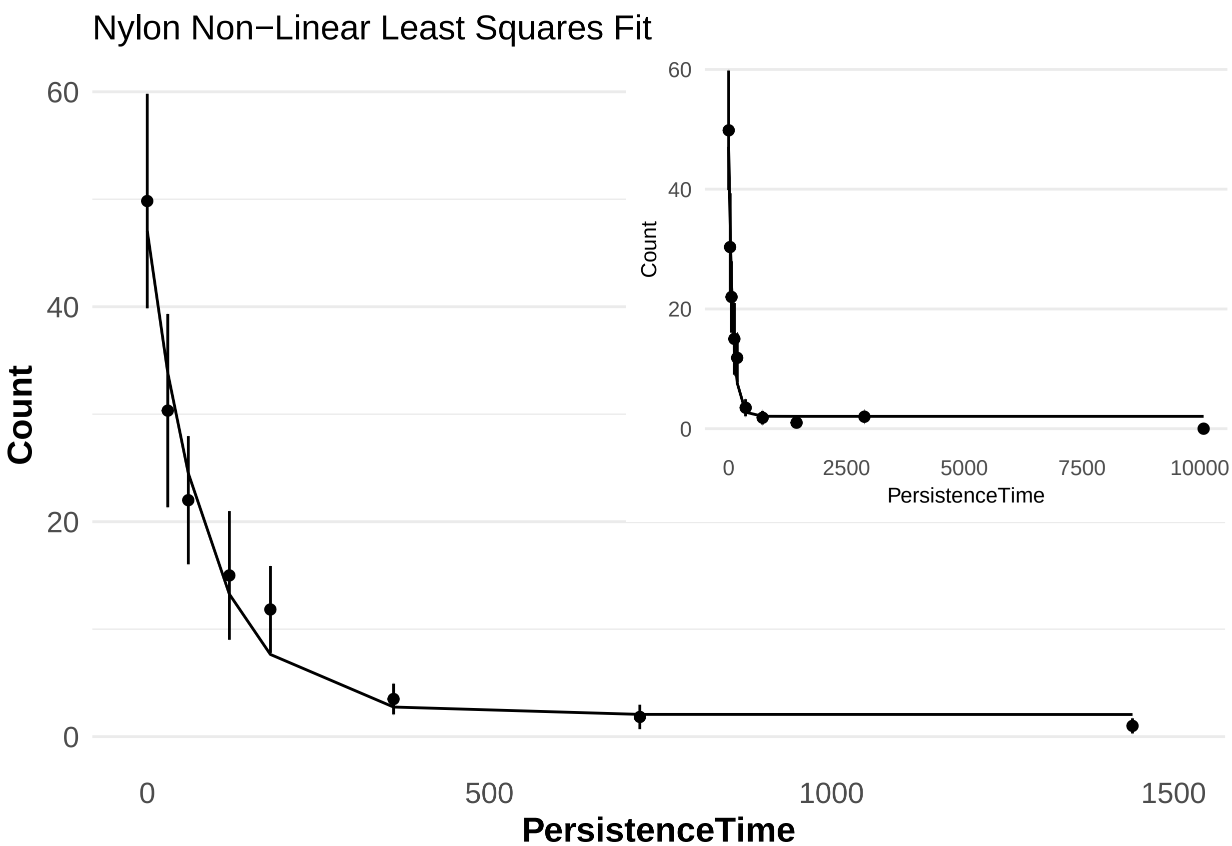
On first impressions the fit looks fine. The residual error is: 6.42217

A plot of the data and the fit will allow a visual comparison.

```r
p1 = ggplot(summ.dat[summ.dat$Substrate == 'Nylon',], aes(x = PersistenceTime, y = Count)) +
  ggtitle("Nylon Non-Linear Least Squares Fit") +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate
  geom_line(aes(y = predict(ny.fit))) +
  xlim(c(-5,1500)) +
  mytheme

p2 = ggplot(summ.dat[summ.dat$Substrate == 'Nylon',], aes(x = PersistenceTime, y = Count)) +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate
  geom_line(aes(y = predict(ny.fit))) +
  inset_theme

p1 + annotation_custom(ggplotGrob(p2), xmin = 700, xmax = 1600,
                       ymin = 20, ymax = 65)
```



The nylon data looks good, let's to the same with the wool data and compare.

```r
wl.fit = nls(Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha),
             data = summ.dat,
             subset = Substrate == 'Wool')
wl.fit.err = summ.dat$Count-predict(wl.fit)
wl.nlm_error <- sqrt(mean(wl.fit.err^2))
wl.fit
```

```
## Nonlinear regression model
##   model: Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha)
```

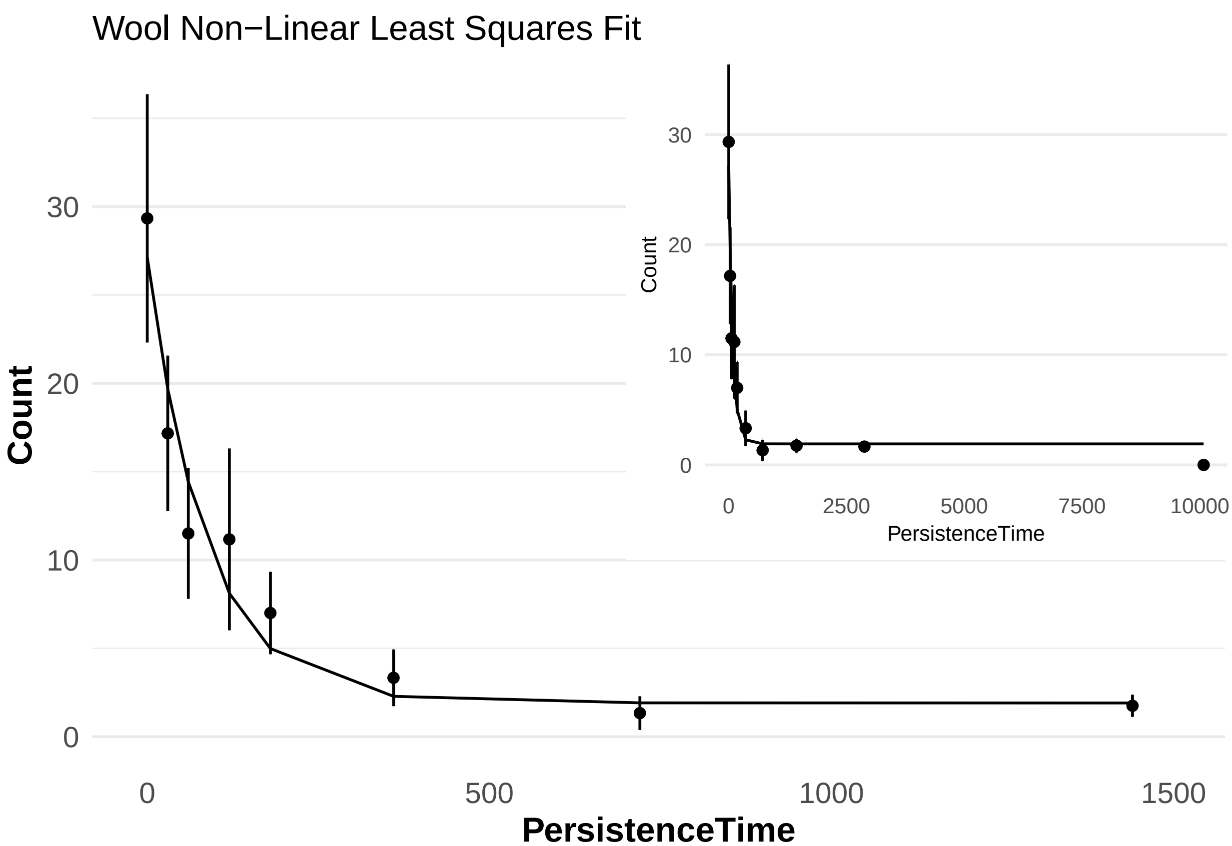
```
##    data: summ.dat
##    Countf     Count0 log_alpha
##     1.908     27.132    -4.449
##  residual sum-of-squares: 38.19
##
## Number of iterations to convergence: 0
## Achieved convergence tolerance: 4.898e-06
```

Residual error: 6.42218

```
p1 = ggplot(summ.dat[summ.dat$Substrate == 'Wool',], aes(x = PersistenceTime, y = Count)) +
  ggtitle("Wool Non-Linear Least Squares Fit") +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate
  geom_line(aes(y = predict(wl.fit))) +
  xlim(c(-5,1500)) +
  mytheme

p2 = ggplot(summ.dat[summ.dat$Substrate == 'Wool',], aes(x = PersistenceTime, y = Count)) +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate
  geom_line(aes(y = predict(wl.fit))) +
  inset_theme

p1 + annotation_custom(ggplotGrob(p2), xmin = 700, xmax = 1600,
                       ymin = 10, ymax = 40)
```




```
cmp.fit.dat = summ.dat %>%
```

```
  group_by(Substrate) %>%
  do(fit = nls(Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha), data = .)) %>%
  tidy(fit) %>%
  select(Substrate, term, estimate) %>%
  spread(term, estimate) %>%
  mutate(alpha = exp(log_alpha))
knitr::kable(cmp.fit.dat)
```

Substrate	Count0	Countf	log_alpha	alpha
Nylon	47.06339	2.060017	-4.457615	0.0115900
Wool	27.13156	1.908363	-4.449466	0.0116848

Interestingly the decay rate (alpha) is almost the same for the two materials.