

# R Notebook

```
froot="D:/sjh/stringmol/smsp/1705smsp/"
mroot="D:/sjh/stringmol/smsp/"

#froot=~/Desktop/paulien/smsp/1705smsp/
#mroot <- ~/Desktop/paulien/smsp/
```

Need a function to stitch the popdys together:

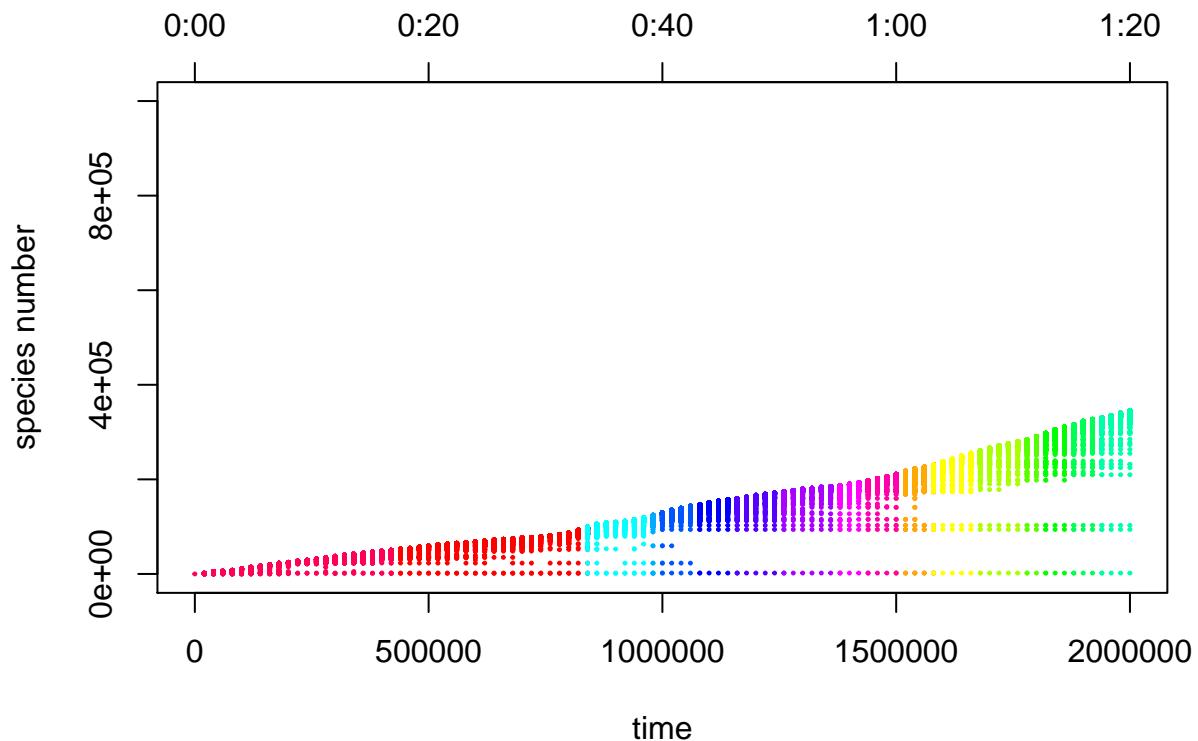
```
# arguments
#function:
stitch.popdy <- function(froot = ".", doplot=F, ylim=c(0,1000000)){
  fl <- list.files(fr, pattern="^popdy*", full.names = T)
  fcols <- rainbow(length(fl))
  datachunk <- list()
  #maxspp = 0
  for(ff in 1:length(fl)){
    #message(sprintf("loading file %d: %s",ff,fl[ff]))
    datachunk[[ff]]<-read.table(fl[ff], col.names=c("time", "spp", "pop"), sep=",", stringsAsFactors = F)

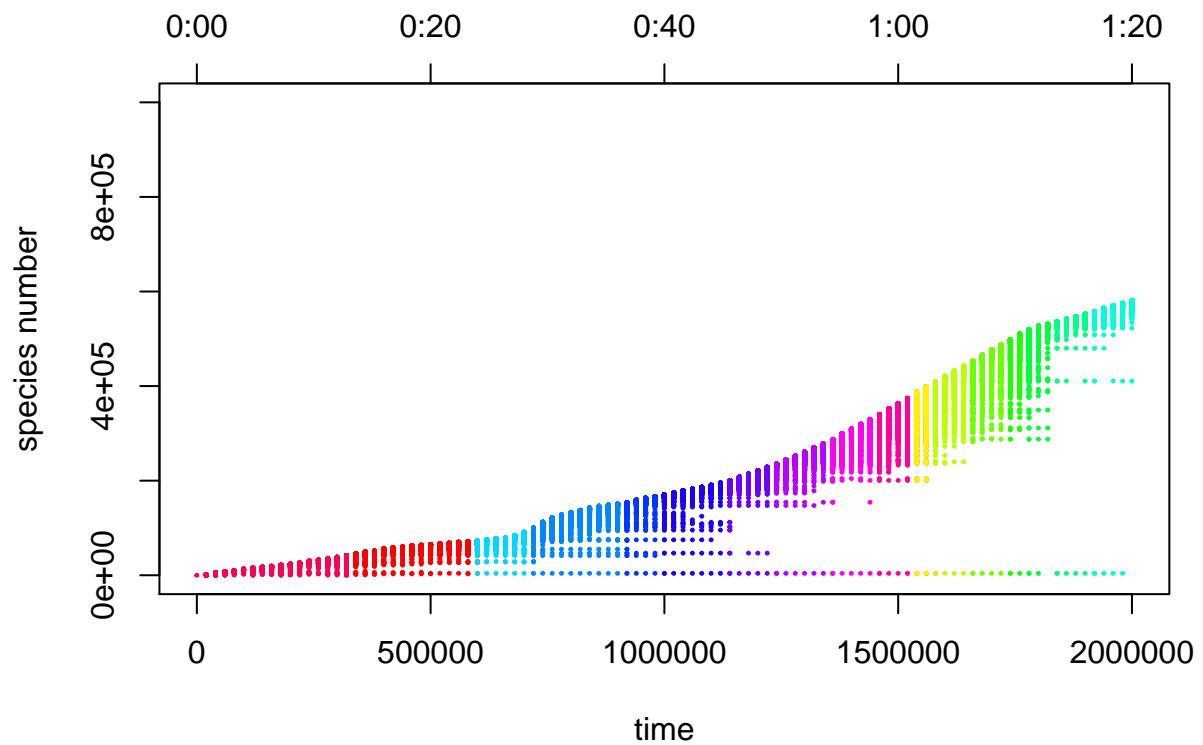
    # maxspp = max(maxspp, max(as.integer(datachunk[[ff]]$spp)))
  }
  #message(sprintf("maxspp is %0.0f",maxspp))
  #maxspp = max(maxspp, ymax)
  for(ff in 1:length(fl)){
    #message(sprintf("plotting file %s",fl[ff]))
    if(ff==1){
      if(doplot){
        plot(x=datachunk[[ff]]$time, y=datachunk[[ff]]$spp, col=fcols[ff], pch=19, cex=0.2, xlim=c(0,2000000))

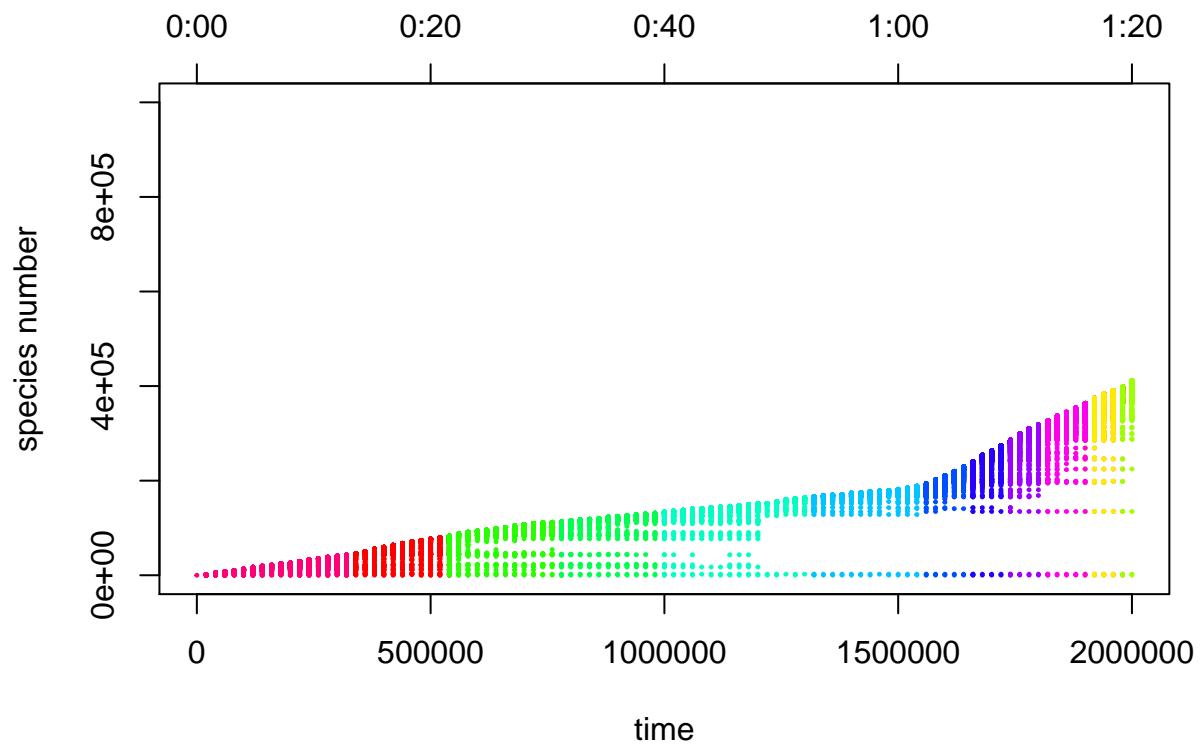
        at <- axTicks(1)
        axis(3, at=at, labels=sprintf("%d:%02d", (at*83/2000000) %% 60, at %% 60), las=1)
      }
      popdy <- data.frame(datachunk[[ff]])
    } else{
      if(doplot){
        points(x=datachunk[[ff]]$time, y=datachunk[[ff]]$spp, col=fcols[ff], pch=19, cex=0.2)
      }
      popdy <- rbind(popdy, data.frame(datachunk[[ff]]))
    }
  }
  popdy <- popdy[order(popdy$time),]
  popdy <- unique(popdy)
  return (popdy)
}

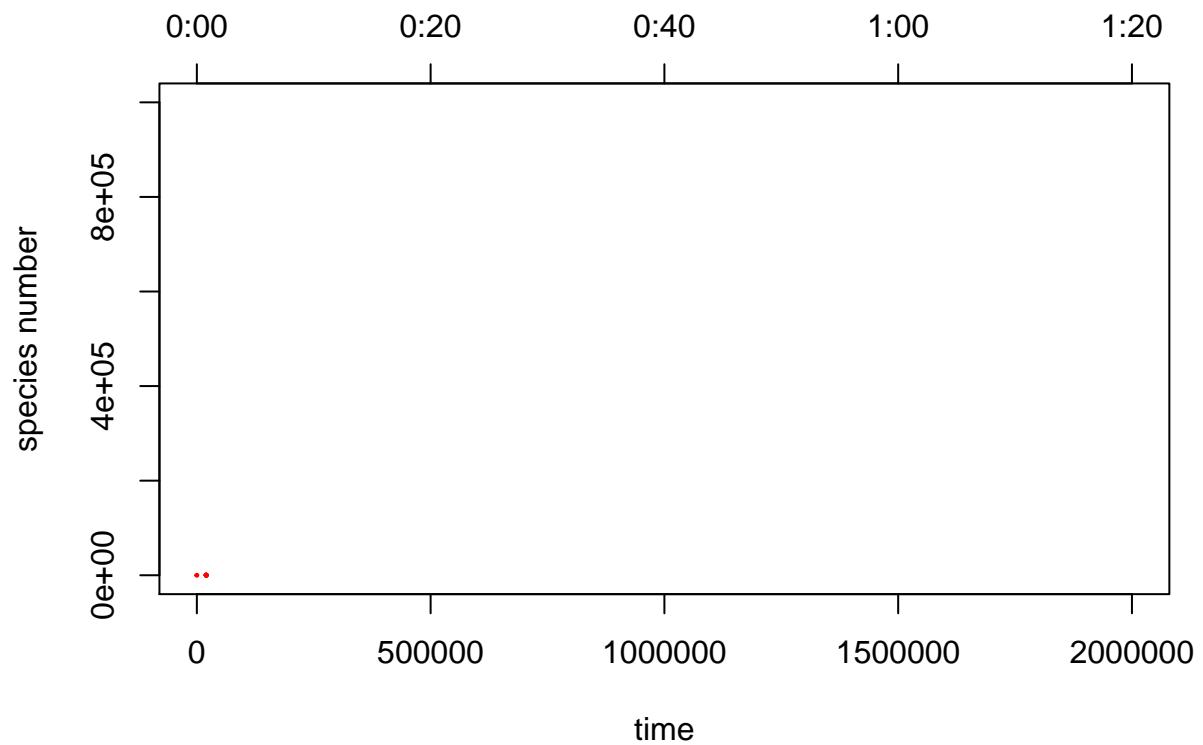
for(rr in 1:5){
  fr = sprintf("%sout%d/", froot, rr)
```

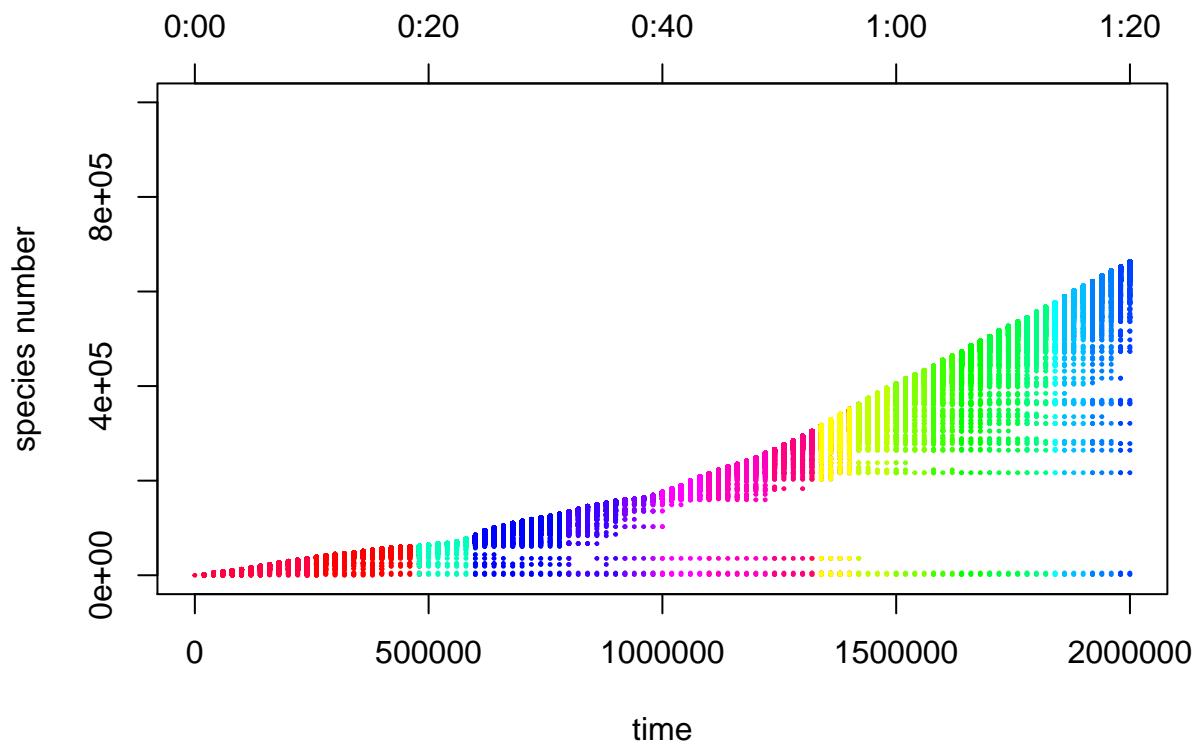
```
    pd <- stitch.popdy(fr,doplot=T)
}
```











```

source("../R/plot_onepop.R")

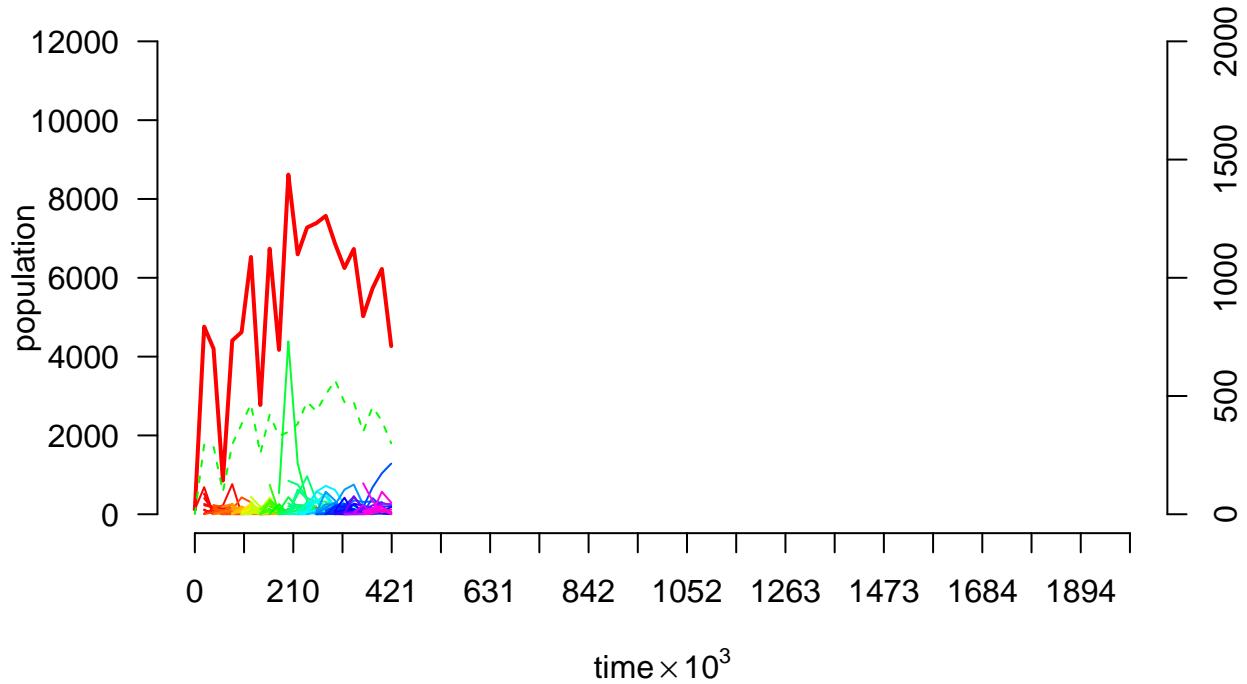
for(tt in 1:5){
fn=sprintf("%sout%d/popdy001.dat",froot,tt)
plot.onepop(fn,sprintf("aexample%dpopdy.pdf",tt),12,12,minpop=100, totpop=T,plotnspp = T,title=sprintf(
})

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

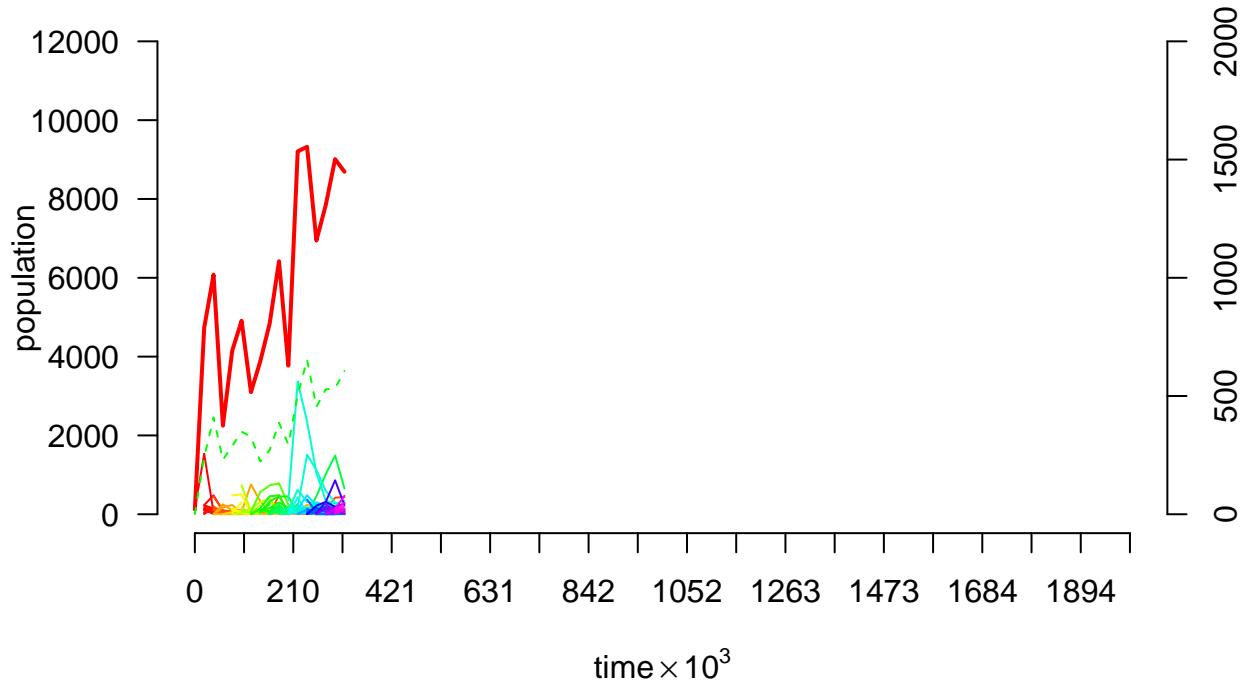
```

## run 1



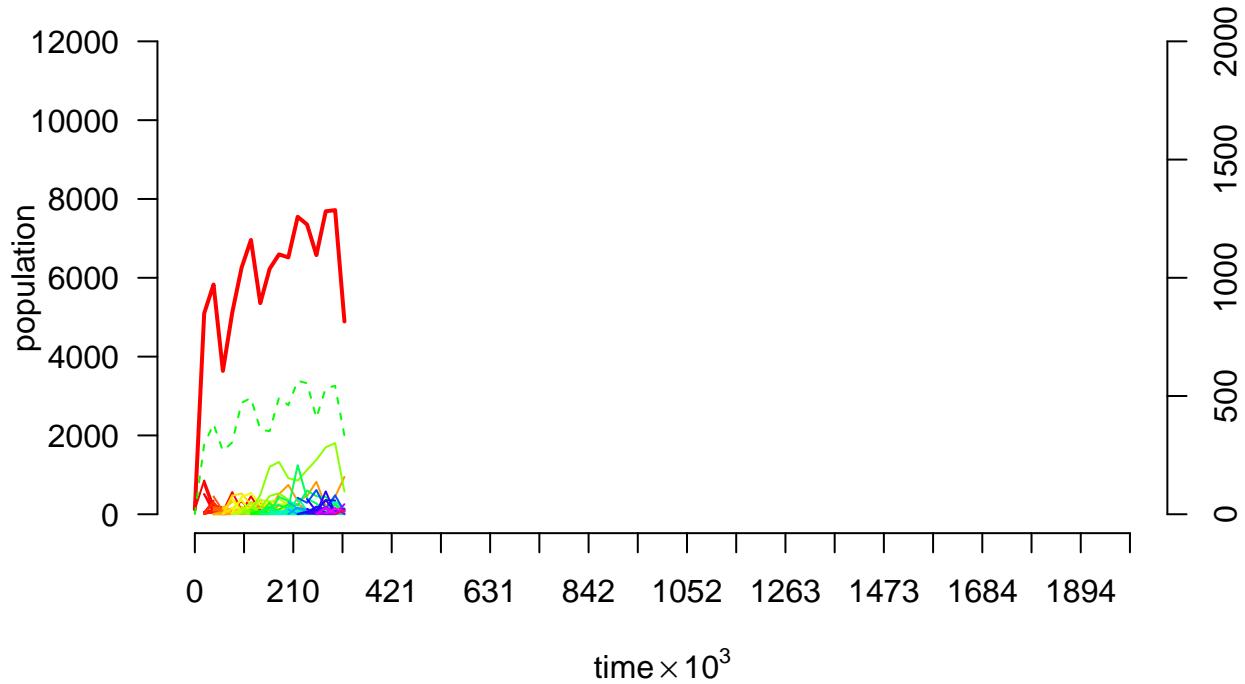
```
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition  
## has length > 1 and only the first element will be used
```

## run 2



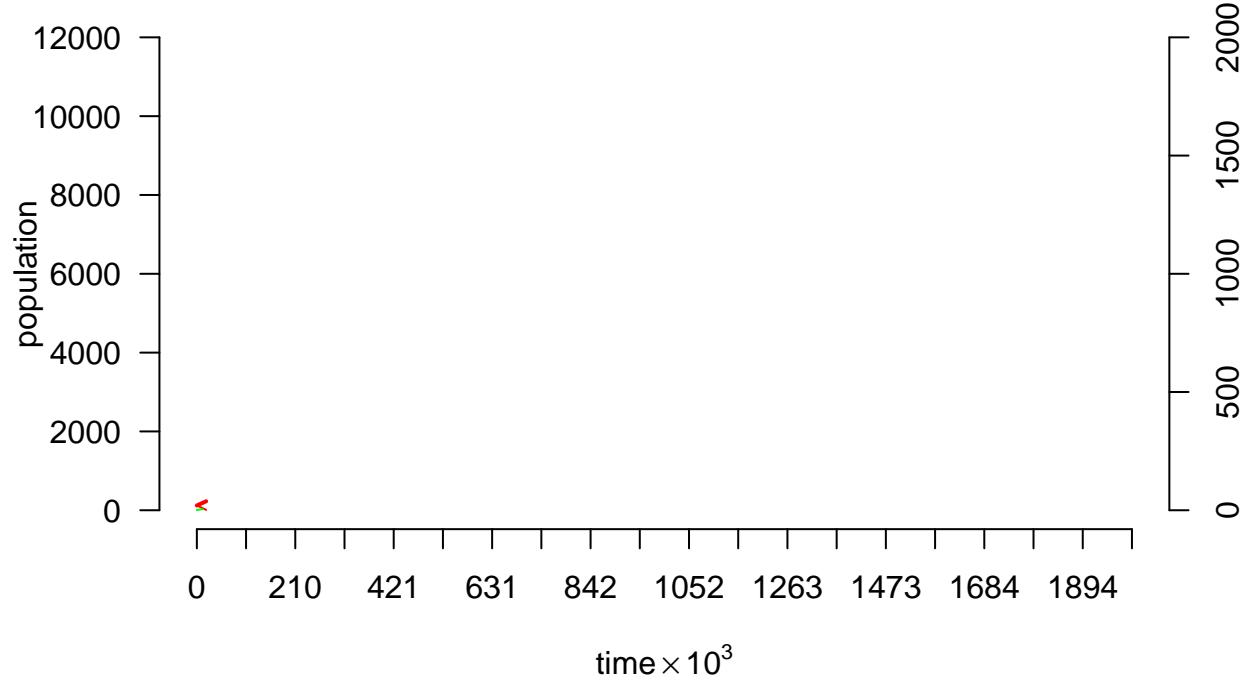
```
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition  
## has length > 1 and only the first element will be used
```

## run 3

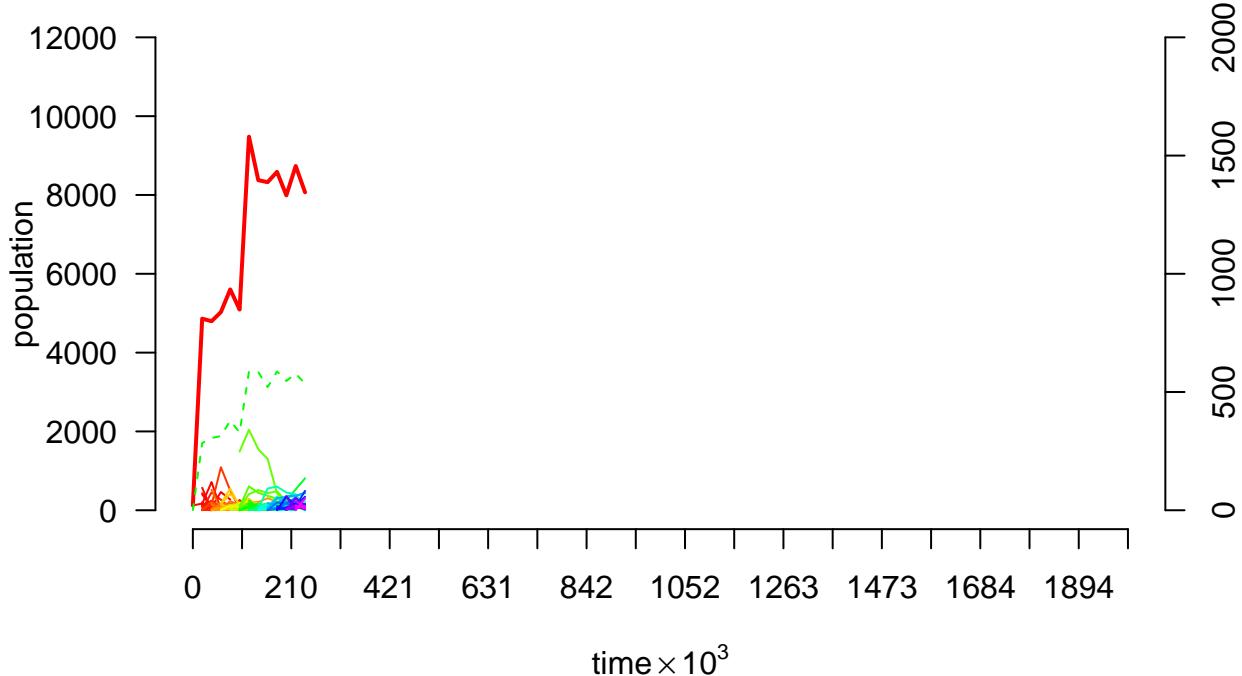


```
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition  
## has length > 1 and only the first element will be used
```

## run 4



## run 5

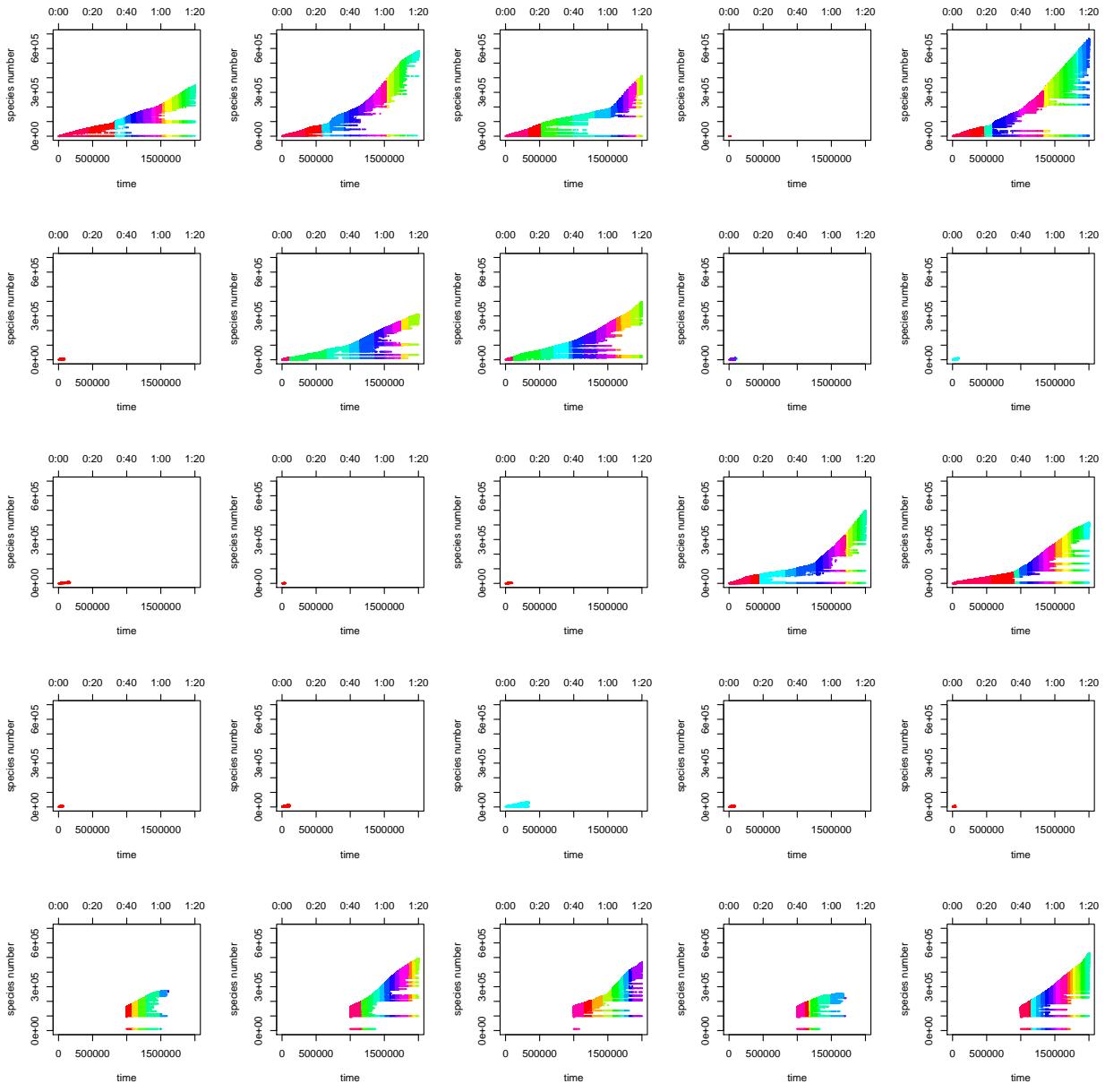


Running this for all samples

```
rundirs <- c("1705smssp", "1705smspr", "1705sm250", "1705sm250r", "evoe
runtypes <- c("125x100, block init", "125x100, rand init", "250x50, block init", "250x50, rand init", "evoe
par(mfrow=c(length(rundirs),5))
for(ss in 1:length(rundirs)){
  for(rr in 1:5){

    fr = sprintf("%s%s/out%d/", mroot, rundirs[ss], rr)
    pd <- stitch.popdy(fr, doplot=T, ylim=c(0,700000))
    write.table(pd, file = sprintf("%s%s/out%d/mpopdy.dat", mroot, rundirs[ss], rr), row.names = F, col.names = F)

  }
}
```



## Popdy stuff

```

require(bacollite)

## Loading required package: bacollite
## Loading required package: stringr
source("../R/plot_onepop.R")
#mroot <- "~/Desktop/paulien/smssp/"
rundirs <- c("1705smssp", "1705smspr", "1705sm250", "1705sm250r", "evoevo")
par(mfrow=c(length(rundirs),5))
for(ss in 1:length(rundirs)){

```

```

for(rr in 1:5){

  fr = sprintf("%s%s/out%d/",mroot,rundirs[ss],rr)

  fn=sprintf("%smpopdy.dat",fr)

  message(sprintf("Processing file %s",fn))

  plot.onepop(fn,sprintf("%sexample%dpopdy.pdf",runtypes[ss],rr),12,12,minpop=100, totpop=T,plotnspp = 100)

}

## Processing file D:/sjh/stringmol/smsp/1705smssp/out1/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smssp/out2/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smssp/out3/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smssp/out4/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smssp/out5/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smspr/out1/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smspr/out2/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smspr/out3/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smspr/out4/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705smspr/out5/mpopdy.dat

```

```

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250/out1/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250/out2/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250/out3/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250/out4/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250/out5/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250r/out1/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250r/out2/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250r/out3/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250r/out4/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/1705sm250r/out5/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/evoevo/out1/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/evoevo/out2/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/evoevo/out3/mpopdy.dat

## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

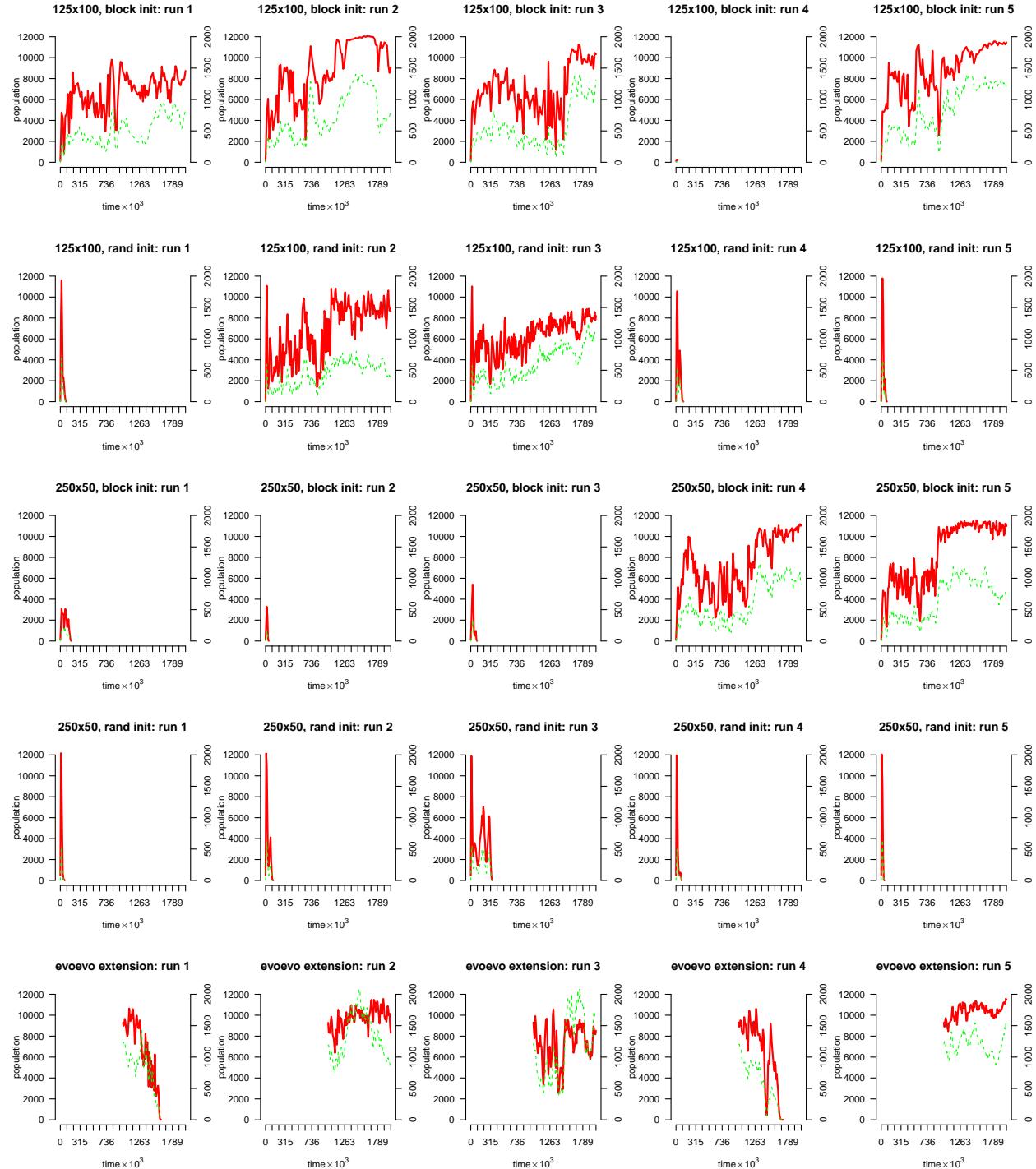
```

```

## Processing file D:/sjh/stringmol/smsp/evoevo/out4/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

## Processing file D:/sjh/stringmol/smsp/evoevo/out5/mpopdy.dat
## Warning in if (is.na(time.lim)) time.lim <- range(x$time): the condition
## has length > 1 and only the first element will be used

```



## removing spurious popdys.

There were a few glitches in running the trials - one of these is that steps 1710000 and 1750000 were run a few times in 1705sm250 trial 5.

These can be found using the following code on the original dataset:

```
rdir <- "../stringmol/smssp/1705sm250/out5/"  
fl <- list.files(rdir,pattern="popdy001*")  
for(ff in fl){  
  pd<-read.table(sprintf("%s%s",rdir,ff),stringsAsFactors = F,sep=",")  
  message(sprintf("%s starts at %d",ff,pd[1,1]))  
}
```

As a result, we can see that it is prudent to prepend the following files with UNUSED:

```
popdy001.13.dat  
popdy001.14.dat  
popdy001.16.dat
```

(Note also that the slist files corresponding to these were overwritten) Once this is done, you just have to re-run `stitch.popdy` to get the master popdy file `mpopdy.dat` for each trial.

## Length Distribution plot

There are three states for molecule: Unbound, active or passive.

```
fn1 <- "~/Desktop/paulien/smssp/1705smssp/out3/out1_900000.conf"  
fn2 <- "~/Desktop/paulien/smssp/1705smssp/out3/out1_1900000.conf"  
source("../R/rconf_data.R")  
source("../R/rconf_stats.R")  
require(stringr)  
  
times <- seq(0,2000000,40000)  
ldata <- list()  
agd <- list()  
acd <- list()  
pad <- list()  
  
for(tt in 2:length(times)){  
  
  fn1<-sprintf("%s/1705sm250/out5/out1_%05d.conf",mroot,times[tt])  
  
  #message(fn1)  
  
  agents1 <- rconf_agents(fn1)  
  eachr <- rconf_rdata(fn1,summarize = F)  
  al <- c(agents1,as.character(eachr$actseq),as.character(eachr$passeq))  
  
  agd[[tt]] <- str_length(agents1)  
  acd[[tt]] <- str_length(eachr$actseq)  
  pad[[tt]] <- str_length(eachr$passeq)  
  
  ldata[[tt]] <- str_length(al)
```

```
}  
par(mfrow=c(4,1))  
boxplot(agd,ylim=c(0,200),pch=19,cex=0.2,main="Length of All stringmols")  
boxplot(acd,ylim=c(0,200),pch=19,cex=0.2,main="Length of Active partners")  
boxplot(pad,ylim=c(0,200),pch=19,cex=0.2,main="Length of Passive partners")  
boxplot(lpdata,ylim=c(0,200),pch=19,cex=0.2,main="Length of all reacting stringmols")
```

