Testing dpf

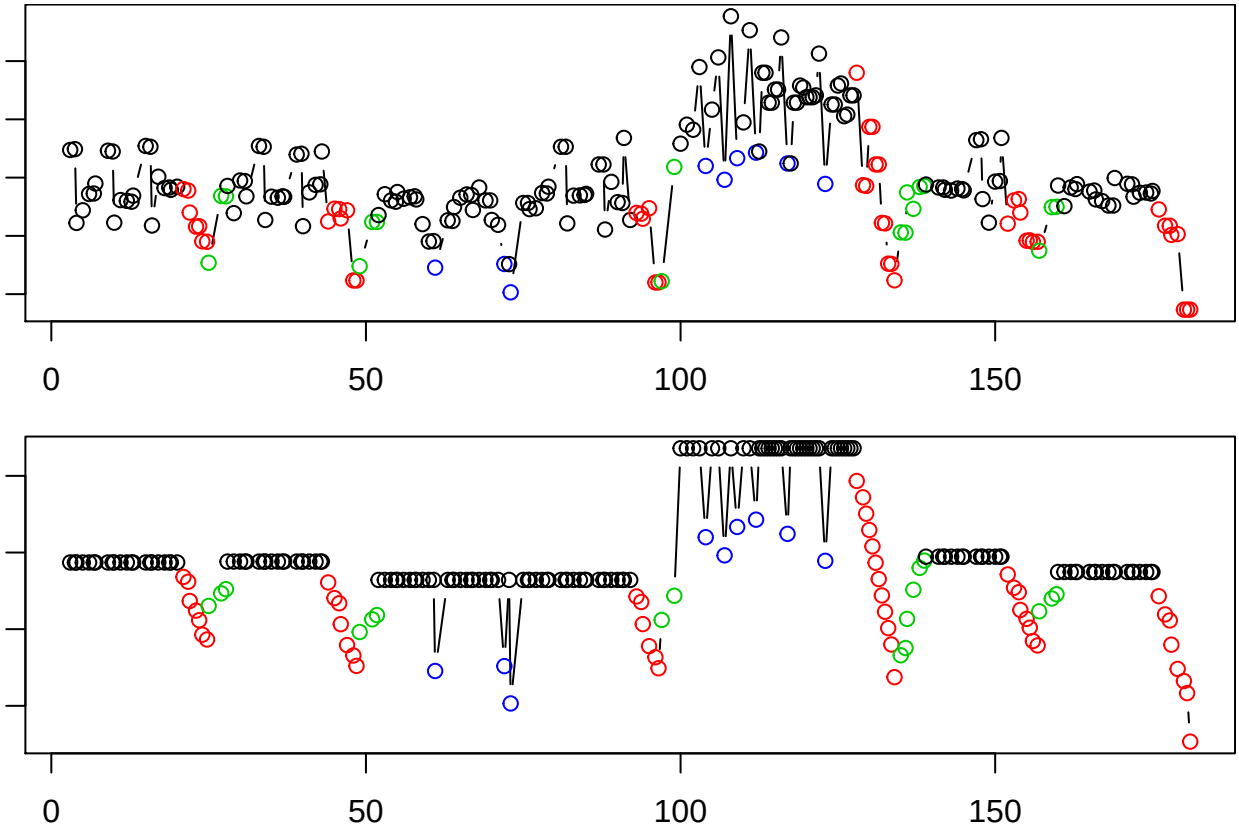
Daniel J. McDonald

2 February 2018

Testing

```
# pull recent dajmcdon/dpf
# devtools::install_github('dajmcdon/dpf')
library(dpf)
```

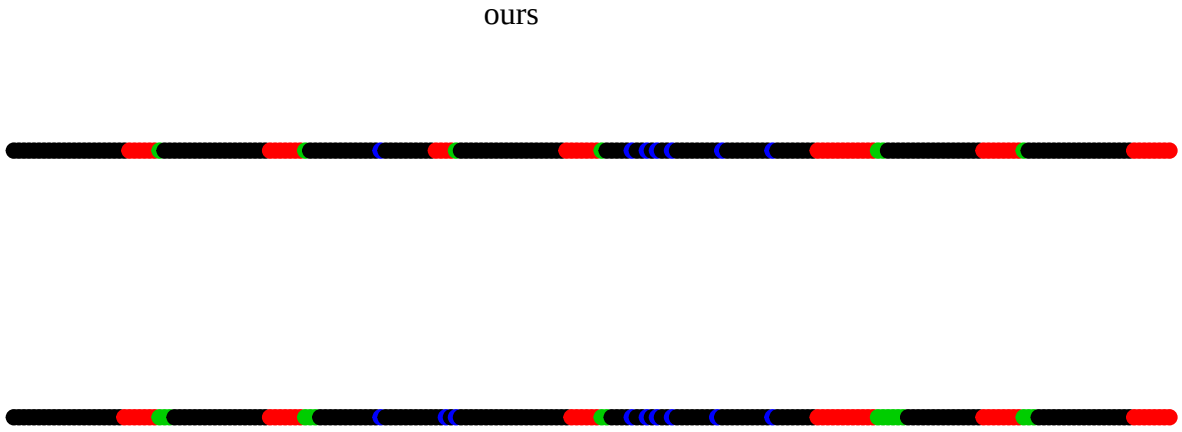
```
load('../music.Rdata')
source('../Beam_Search.R') ## Yupeng's original beam search on some music data
```



This plots our discrete states and his.

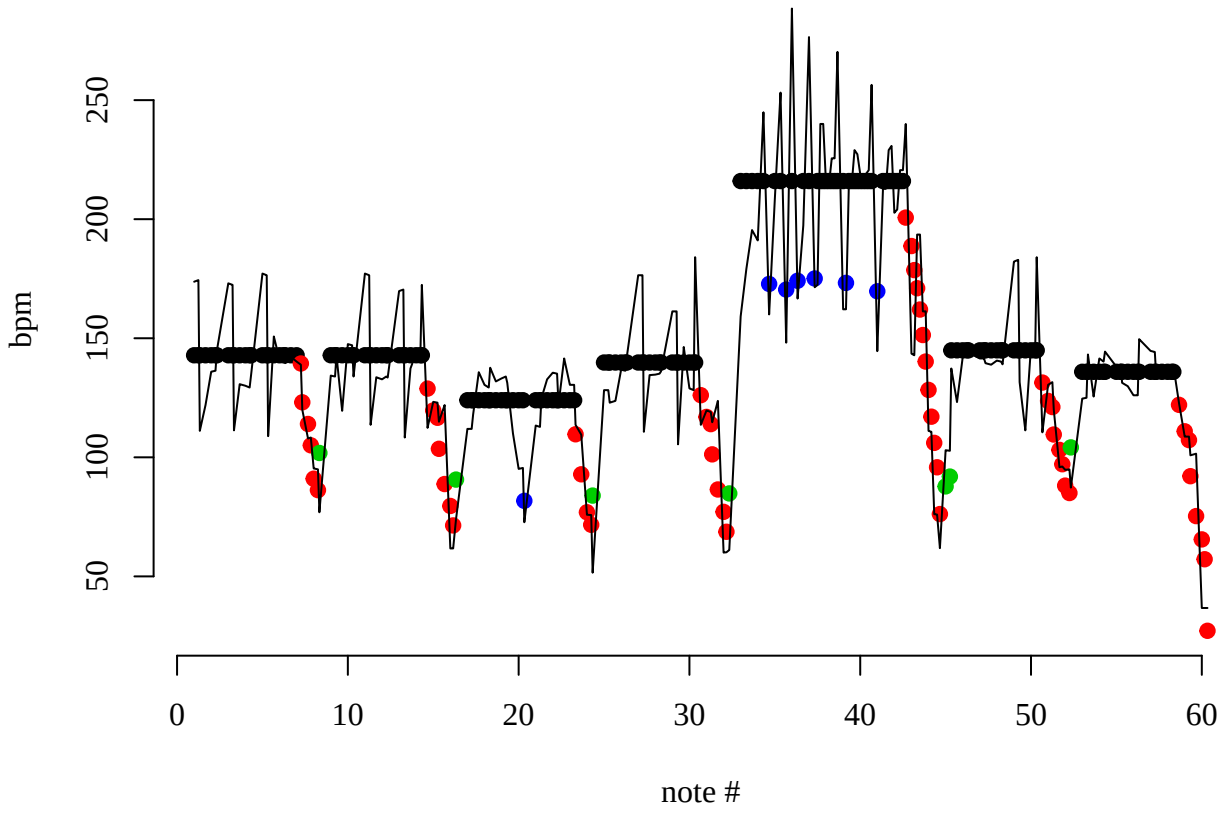
```
lt = ioi
out = yupengMats(lt, sd_l^2, c(mean(y),-40,-40), c(.01,20,10,30)^2, c(.8,.1,.8,.4))
test = beamSearch(out$a0,out$P0,c(1,0,0,0,0,0,0,0), out$dt, out$ct, out$Tt, out$Zt,
out$Rt, out$Qt, out$GGt, matrix(y,nrow = 1), out$transMat, 50)
bestpath = test$paths[which.max(test$weights),]
par(mfrow=c(1,1))
par(mar=c(0,0,0,0),family='serif')
nn = length(y)
plot(1:nn,rep(1,nn),col=convert8to4(bestpath),
     pch=19,ylim=c(-1,2),bty='n',xlab='',ylab='',yaxt='n',xaxt='n')
```

```
points(1:nn, rep(0,nn), col=s, pch=19)
text(100,1.5,"ours")
text(100,-.5,"yupeng's")
```



This plots our continuous states.

```
filt = pathStuff(out, bestpath, matrix(y,nrow=1))
par(mar=c(5,4,0,0),family='serif')
plot(note.onset,filt$ests,col=convert8to4(bestpath),
     pch=19,bty='n',xlab='note #',ylab='bpm',ylim=range(y,filt$ests))
lines(note.onset, y)
```


bpm
50
100
150
200
250
0
10
20
30
40
50
60
note #