R-code for producing Figure 2 from Pedersen et al. (2013), Nature Biotechnology

Lykke Pedersen, Peter H Hagedorn, Marie Lindholm, Morten Lindow

With the gain of increasing reproducibility, this vignette includes the commands to reproduce Fig. 2 from "A kinetic model of enzyme recruiting oligonucleotides predicts an optimal affinity and thus explains why shorter and less affine oligonucleotides may be more potent". The R-functions from the ASOmodels are used and the package is loaded by the commands

```
> require(devtools)
> install_github('ASOmodel',username='lykkep')
> require(ASOmodels)
```

Kinetic model figures

Figure 2a: Time-resolved simulation of the ASO model

Parameters for the ASO model, the initial concentrations and the time-steps for which the simulation is performed:

```
> parms <- c(Et = 1,KdOT = 0.3,kOpT = 0.2,KdOTE = 70,kOTpE = 5,
+            vprod = 0.2,vdegrad = 0.04,alpha=0.1,kcleav = 8)
> init <- c(T=parms['vprod']/parms['vdegrad'], OT=0, OTE=0,
+           E=parms['Et'], O=100, OCE=0, OC=0)
> TimeSteps <- c(seq(0,4.3,by=5E-2),seq(5,65,by=1))
```

Using `vode()` the ASO model is simulated in time. The function `diffASO()` is part of the ASOmodels package.

```
> solASO <- vode(init,TimeSteps,diffASO,parms)
```

The timetraces for the concentrations of $[O]$, $[T]$, $[OT]$, $[OTE]$, and $[E]$ are plotted:

```
> SSvalue <- signif(last(solASO)[-1],2)
> solASO <- apply(solASO[,2:8],2,
+                 function(x) (x-min(x))/max(x-min(x)) )
> colVAR <- c('black','darkgreen','darkred','orange','green')
```

```
> xtime <- TimeSteps <= 35
> par(mar=c(3.2,3.4,0.1,0.1),bty='n',mgp=c(2,0.7,0),
+     cex=0.6,cex.axis=1,las=1)
> for(i in 1:5){
+   if(i!=1) par(new=TRUE)
+   plot(TimeSteps[xtime], solASO[xtime,i], yaxt='n', xaxt='n',
+        ylab='relative concentrations', xlab='minutes',las=1,
+        col=colVAR[i], type='l', ylim=c(0,1), xlim=c(0,35+26))
+ }
> xtime <- 40
> for(i in 1:5) lines(xtime+0:20,rep(last(solASO)[i],21),
+                     col=colVAR[i])
> axis(1,at=c((0:3)*10,45),label=c((0:3)*10,''))
> axis(1,at=45,label='steady-\nstate',mgp=c(0,1.6,0))
> axis(2,at=c(0,1),label=c('min','max'),las=1)
> #O
> text(xtime,last(solASO)[5]-0.05,col=colVAR[5],adj=0,
+      substitute(O == e~nM,list(e=SSvalue[5])))
> #T
> text(xtime,0.05,col=colVAR[1],adj=0,
+      substitute(T == e*pM,list(e=1e3*SSvalue[1])))
> #OT
> text(xtime,last(solASO)[2]-0.05,col=colVAR[2],adj=0,
+      substitute(OT== e*nM,list(e=SSvalue[2])))
> #OTE
> text(xtime,last(solASO)[3]+0.05,col=colVAR[3],adj=0,
+      substitute(OTE == e*pM,list(e=1e3*SSvalue[3])))
> #E
> text(xtime,last(solASO)[4]+0.05,col=colVAR[4],adj=0,
+      substitute(E == e*nM,list(e=SSvalue[4])))
```

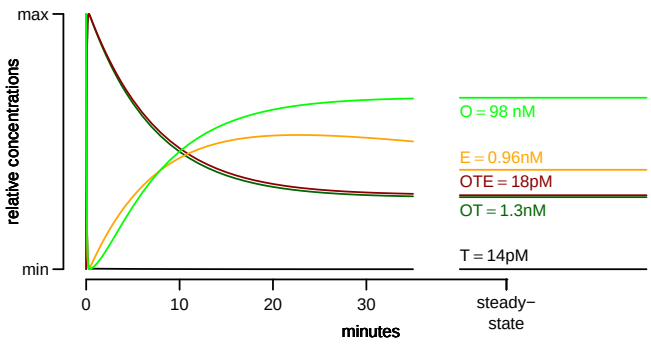

Figure 2a: Time resolved simulation of the relative concentrations of key species

Figure 2b: Simulated dose-response curve

Given a set of parameters the R-function `Trel()` from the ASOmodels package calculates the relative target concentration as a function of the total concentration of oligonucleotide added to the system.

```
> curve(Trel,1E-3,5E2,log='x', lwd=2,ylim=c(0,1),
+       ylab=expression(T[rel]),xaxt='n',
+       xlab='Total oligonucleotide conc (nM)')
> abline(h=Trel(1E9),lty=2) #Trel,min
> abline(v=IC50(parms['KdOT']),lty=2) #IC50
> axis(1,at=10^c(-3,-1,1,3),
+      labels=pretty10expLP(10^c(-3,-1,1,3),drop.1=T))
> axis(1,at=IC50(parms['KdOT']),label=expression(IC[50]))
> axis(2,at=Trel(1E6),
+      label=expression(T[rel*','*min]),las=1)
```

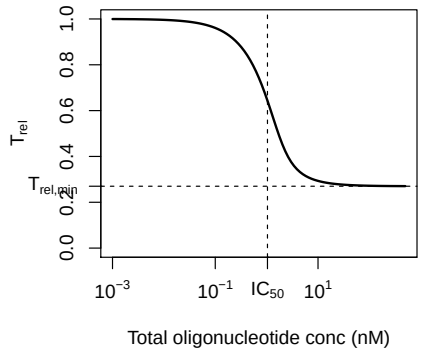

Figure 2b: The relative total target concentration (T_{rel}) is defined as the steady state level of total target in the presence of oligonucleotide divided by the target concentration in the absence of oligonucleotide. Dashed lines indicate efficacy (horizontal) and IC_{50} (vertical).

Figure 2c: An optimum affinity

For a sequence of affinities `D1_seq` the IC_{50} values are calculated by use of the R-function `IC50` from the ASOmodels package:

```
> D1_seq <- 10^seq(-3,3.2,by=0.25)
> ICfit <- sapply(D1_seq,IC50)
```

When there is no coupling between the off-rates $k_{OT\to O+T}$ and $k_{*C\to *+C}$ then the value of $k_{*C\to *+C}$ is set in the param vector as the entry `'kC'`:

```
> parmsNO <- c(parms,kC=parms['kOpT']*parms['KdOT']/parms['alpha'])
> names(parmsNO)[length(parmsNO)] <- 'kC'
> ICfitNO <- sapply(D1_seq,IC50NO) #IC50 without coupling
```

For the sequence of affinities the sequences of IC_{50} values are plotted:

```
> plot(D1_seq,ICfit,log='xy',yaxt='n',type='l',xaxt='n',
+      xlab=expression(D[OT]~'(nM)'),ylab=expression(IC[50]~'(nM)'))
> lines(D1_seq,ICfitNO,lty=2)
> axis(2,at=c(2,20,200),labels=c(2,20,200))
> axis(1,at=10^pretty(log10(D1_seq)),
+      labels=pretty10expLP(10^pretty(log10(D1_seq)),drop.1=T),)
> legend('topleft',c('Coupling','No coupling'),lty=c(1,2),bty='n')
```

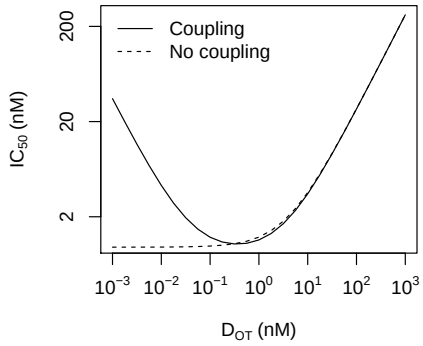

Figure 2c: The IC_{50} as a function of the dissociation constant for the OT complex. A low K_{dOT} corresponds to a high affinity binding. Dashed line: no coupling of off-rates. Solid line: coupling of off-rates.

Experimental data figures

Figure 2d: Frieden et al. (2003)

```
> data(gapmers)
> dat <- data.frame(gapmers)
> colL <- c('red','orange','darkgreen','','darkblue','','purple','','black')
> #### We plot the data from Frieden et al, 2003
> dat.F <- dat[dat$Study=="Frieden 2003",]
> cohigh.F <- 63; colow.F <- 53
> tmp <- abs(cohigh.F-colow.F)
> cut.F <- cut(dat.F$Predicted.Tm,c(0,colow.F,cohigh.F,100),labels=F)
```

```
> Fx <- lapply(1:3,function(i)dat.F$Predicted.Tm[cut.F==i])
> Fy <- lapply(1:3,function(i) dat.F$Dose.2nm[cut.F==i])
> Flength <- dat.F$Oligo.length
> bp <- barplot(sapply(Fy,mean),ylim=c(0,55), las=1, axes=F,
+               yaxs='i',xaxs='i',space=0.01)
> plotCI(bp[,1],sapply(Fy,mean),sapply(Fy,sd),
+        add=T,pch=NA, gap=0,yaxs='i')
> par(new=T)
> plot(unlist(Fx),unlist(Fy),xlim=c(colow.F-tmp,cohigh.F+tmp),
+      ylim=c(0,55), pch=19, col=colL[Flength-11],xaxt='n',
+      ylab='% target measured from luciferase',
+      xlab=expression(T[m]~'('*degree*C*')'),yaxs='i',xaxs='i')
> axis(1,at=c(colow.F,cohigh.F),labels=as.character(c(colow.F,cohigh.F)))
> legend('bottomright',as.character(sort(unique(Flength))),
+        pch=19, col=colL[sort(unique(Flength))-11],bg='white')
```

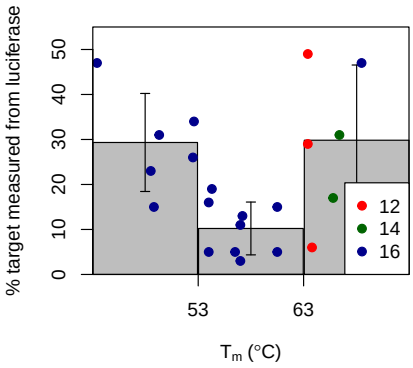

Figure 2d: 21 oligonucleotides targeted against the luciferase firefly gene.

Figure 2e: Stanton et al. (2012)

```
> #### We plot the data from Stanton et al 2012
> dat.S <- dat[dat[,1]=="Stanton 2012",]
> cohigh.S <- 61; colow.S <- 46
> tmp <- abs(cohigh.S-colow.S)
> cut.S <- cut(dat.S$Predicted.Tm,c(0,colow.S,cohigh.S,100),labels=F)
> Sx <- lapply(1:3,function(i)dat.S$Predicted.Tm[cut.S==i])
> Sy <- lapply(1:3,function(i) dat.S$Dose.3nm[cut.S==i])
> Slength <- dat.S$Oligo.length
> bp <- barplot(sapply(Sy,mean),ylim=c(0,105), las=1,axes=F,
```

```
+               yaxs='i', xaxs='i',space=0.01)
> plotCI(bp[,1],sapply(Sy,mean),sapply(Sy,sd),add=T,pch=NA, gap=0,yaxs='i')
> par(new=T)
> plot(unlist(Sx),unlist(Sy),xlim=c(colow.S-tmp,cohigh.S+tmp),
+      ylim=c(0,105), pch=19,col=colL[Slength-11],xaxt='n',
+      ylab='% target measured from PCR',
+      xlab=expression(T[m]~'('*degree*C*')'),yaxs='i',xaxs='i')
> axis(1,at=c(colow.S,cohigh.S),labels=as.character(c(colow.S,cohigh.S)))
> legend('bottomright',as.character(sort(unique(Slength))),
+        pch=19,col=colL[sort(unique(Slength))-11],bg='white')
```

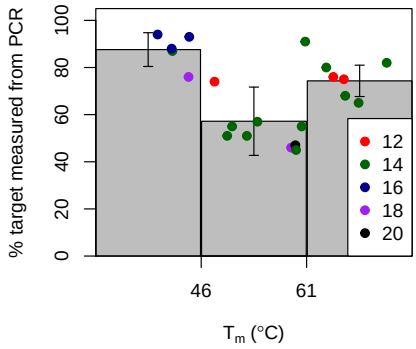

Figure 2e: 21 oligonucleotides targeted against the glucocorticoid receptor.

Figure 2f: Pedersen et al. (2013) (this work)

```
> ### We plot the data from Pedersen et al, 2013
> dat.P <- dat[dat$Study=="Pedersen 2013",]
> cohigh.P <- 56; colow.P <- 47
> tmp <- abs(cohigh.P-colow.P)
> cut.P <- cut(dat.P$Predicted.Tm,c(0,colow.P,cohigh.P,100),labels=F)
> Px <- lapply(1:3,function(i)dat.P$Predicted.Tm[cut.P==i])
> Py <- lapply(1:3,function(i) dat.P$IC50[cut.P==i])
> Plength <- dat.P$Oligo.length
> bp <- barplot(sapply(Py,mean),ylim=c(0,0.016), las=1,
+               axes=F,yaxs='i', xaxs='i',space=0.01)
> plotCI(bp[,1],sapply(Py,mean),sapply(Py,sd),add=T,
+        pch=NA, gap=0,yaxs='i')
> par(new=T)
> plot(unlist(Px),unlist(Py),xlim=c(colow.P-tmp,cohigh.P+tmp),ylim=c(0,0.016),
```

```
+      pch=19,col=colL[Plength-11],xaxt='n',ylab=expression(IC[50]~'('*nM*')'),
+      xlab=expression(T[m]~'('*degree*C*')'),yaxs='i',xaxs='i')
> axis(1,at=c(colow.P,cohigh.P),labels=as.character(c(colow.P,cohigh.P)))
> legend('bottomright',as.character(sort(unique(Plength))),
+        pch=19,col=colL[sort(unique(Plength))-11],bg='white')
```

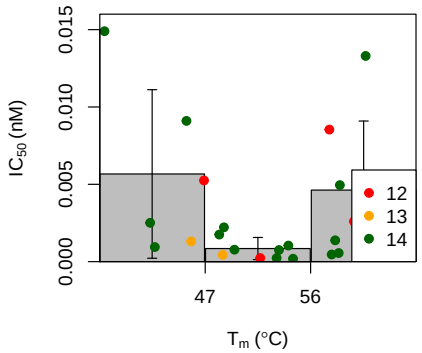

Figure 2f: 23 oligonucleotides targeted against ApoB.