Supplementary **ASOmodel**

Lykke Pedersen, Peter H Hagedorn, ... and Morten Lindow

April 25, 2013

This document is the supplementary file for the manuscript entitled "A kinetic model of enzyme recruiting oligonucleotides predicts an optimal affinity and thus explains why shorter and less affine oligonucleotides may be more potent" (2013) and a vignette for the R-package ASOmodel.

The functions and data used to produce the figures in the main manuscript and this supplementary file are available after installing and requiring the ASOmodel package in R

```
> system('R CMD install ~/Documents/ASOmodels/ASOmodels')
> require(ASOmodels)
```

1 Steady-state of the ASO model

The kinetic ASO model governs eight ODEs for the seven variables: free target (T), free oligonucleotide (O), free RNAse H (E), complex of oligonucleotide and target (OT), complex of oligonucleotide, target and RNAse H (OTE), complex of cleaved target, oligonucleotide and RNase H (OCE), and complex of cleaved target and oligonucleotide (OC). The ASO model is described by the eight equations

$$\frac{\mathrm{d}[T]}{\mathrm{d}t} = v_{prod} - v_{degrad}[T] - k_{O+T\to OT}[T][O] + k_{OT\to O+T}[OT] \tag{1}$$

$$\begin{aligned}\frac{\mathrm{d}[OT]}{\mathrm{d}t} &= k_{O+T\to OT}[O][T] - k_{OT\to O+T}[OT] \\ &\quad - k_{OT+E\to OTE}[OT][E] + k_{OTE\to OT+E}[OTE] - v_{degrad}[OT]\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{\mathrm{d}[OTE]}{\mathrm{d}t} &= k_{OT+E\to OTE}[E][OT] - k_{OTE\to OT+E}[OTE] \\ &\quad - (v_{degrad} + k_{OTE\to OCE})[OTE]\end{aligned} \tag{3}$$

$$\begin{aligned}\frac{\mathrm{d}[E]}{\mathrm{d}t} &= -k_{OT+E\to OTE}[E][OT] + k_{OTE\to OT+E}([OTE] + [OCE]) \\ &\quad + v_{degrad}[OTE]\end{aligned} \tag{4}$$

$$\frac{\mathrm{d}[O]}{\mathrm{d}t} = k_{OT\to O+T}[OT] - k_{O+T\to OT}[O][T] \\ + k_{*C\to *+C}[OC] + v_{degrad}([OT] + [OTE]) \tag{5}$$

$$\frac{\mathrm{d}[OCE]}{\mathrm{d}t} = k_{OTE\to OCE}[OTE] - k_{OTE\to OT+E}[OCE] - k_{*C\to *+C}[OCE] \tag{6}$$

$$\frac{\mathrm{d}[OC]}{\mathrm{d}t} = k_{OTE\to OT+E}[OCE] - k_{*C\to *+C}[OC] \tag{7}$$

Complex formation and breaking are denoted by rate constants k with subscripts. Target production and degradation rates are denoted by v_{prod} and v_{degrad}, respectively. The default parameter values are listed in Table S1.

Steady-state is reached when the Eqs. (1)-(8) are equated to zero. Using Maple16 the steady-state concentrations are found. They all depend on the roots to a fourth order polynomial with coefficients calculated within the Trel.function(). The one root that ensures that all concentrations are non-negative and also fullfills that

$$[O] + [OTE] + [OT] + [OCE] + [OC] \leq O_t \quad \text{and} \\ [OTE] + [OCE] + [E] \leq E_t \ ,$$

is chosen.

Supplementary Table S1: Default values for the parameter-space of the AON model. (1): Amirkhanov and Chattopadhyaya (2002), J. Chem. Soc. (2): Christensen et al (2001), Biochem J. (3): Yang et al. (2003) Genome Res. (4): Cell Biology. REMEMBER TO CHECK THE VALUES!!!

Parameter	Description	Default value	Ref
E_t	Total RNAse H conc	1 nM	(1)
O_t	Total oligo concentration	$\mathcal{O}(\mu M)$	
v_{prod}	Production of target	0.2 nM/min	(4)
v_{degrad}	Degradation of target	0.04 min^{-1}	(3)
D_{OT}	Dissociation constant of OT	0.3 nM	(2)
D_{OTE}	Dissociation constant of OTE	70 nM	(1)
$k_{O+T\to OT}$	Rate of $O + T \to OT$	0.2 (nM min)$^{-1}$	(2)
$k_{OT+E\to OTE}$	Rate of $OT + E \to OTE$	5 (nM min)$^{-1}$	(1)
$k_{OTE\to OCE}$	Rate of $OTE \to OCE$	8 min^{-1}	(1)
α	Ratio of $\frac{k_{OT\to O+T}}{k_{*\mathrm{C}\to *+\mathrm{C}}} \leq 1$	0.1	

When there is no oligonucleotide added to the system, then the steady-state concentration of target is $[T] = \frac{v_{prod}}{v_{degrad}}$. When oligonucleotide is added to the system then the total concentration of target at steady-state is the sum of the concentrations $[T]$, $[OT]$ and $[OTE]$. The relative total target concentration at steady-state is then calculated as

$$T_{\text{rel}} = \frac{[T] + [OT] + [OTE]}{\frac{v_{prod}}{v_{degrad}}} \quad . \tag{8}$$

The half maximal inhibitory concentration (IC_{50}) is the concentration of total nucleotide needed to inhibit the target concentration by half. The IC_{50} is a measure of the potency of an oligonucleotide. A more potent oligonucleotide will have a lower IC_{50} value. In mathematical terms the IC_{50} value is defined as

$$IC_{50} = \left(O_t \,\middle|\, T_{\text{rel}} = \frac{\text{Eff}}{2} + T_{\text{rel,min}} \right) \quad , \tag{9}$$

where the efficacy (Eff, the maximum decrease in T_{rel}), and the minimum value of T_{rel} ($T_{\text{rel,min}}$) are defined by

$$\text{Eff} = 1 - \lim_{O_t \to \infty} T_{\text{rel}} = 1 - T_{\text{rel,min}} \quad . \tag{10}$$

1.1 R-functions to calculate T_{rel} and IC_{50}

The R-function Trel.function() calculates T_{rel} and takes O_t and the set of parameters as input

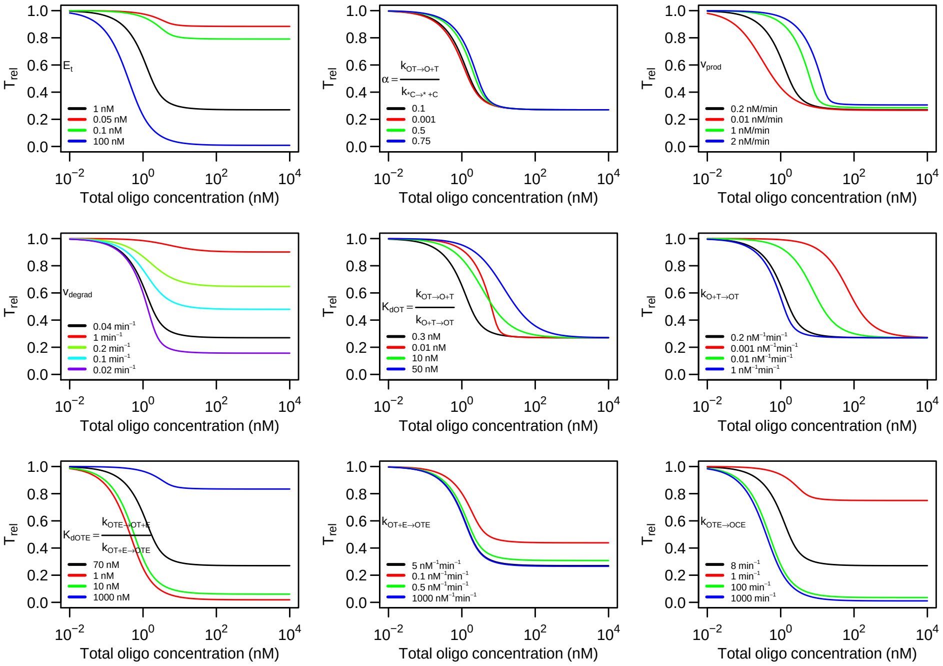
```
> parms <- c(Et = 1,KdOT = 0.3,kOpT = 0.2,KdOTE = 70,kOTpE = 5,
+             vprod = 0.2,vdegrad = 0.04,alpha=0.1,kcleav = 8)
> Trel.function(otot=1,param=parms)
```

```
[1] 0.6538694
```

For a sequence of different oligonucleotide concentrations (O_t), T_{rel} can be calculated for varying parameter values. From this a dose-response curve is obtained. Figure S1 shows the change in the dose-reponse curve as the parameters vary. The plots in Fig. S1 are produced using the change.par() function, e.g., setting the total RNAse H concentration E_t to 0.01 and 1 is done by running the command

```
> change.par('Et',c(0.01,1),
+             expression(Total~RNAse~H~E[t]~'(nM)'),plot=F)
```

Using the R-function drm() from the drc package (v2.3-0) a dose-response curve is fitted to T_{rel} as a function of O_t to obtain an IC_{50} value. This is calculated through the function IC50() that takes D_{OT} and the set of parameters as input:



Supplementary Figure S1: Dose-response curves for different values of E_t, α, v_{prod}, v_{degrad}, D_{OT}, $k_{O+T\to OT}$, D_{OTE}, $k_{OT+E\to OTE}$, and $k_{OTE\to OCE}$ (top,left to bottom,right). Black lines correspond to the parameter values listed in Table S1.

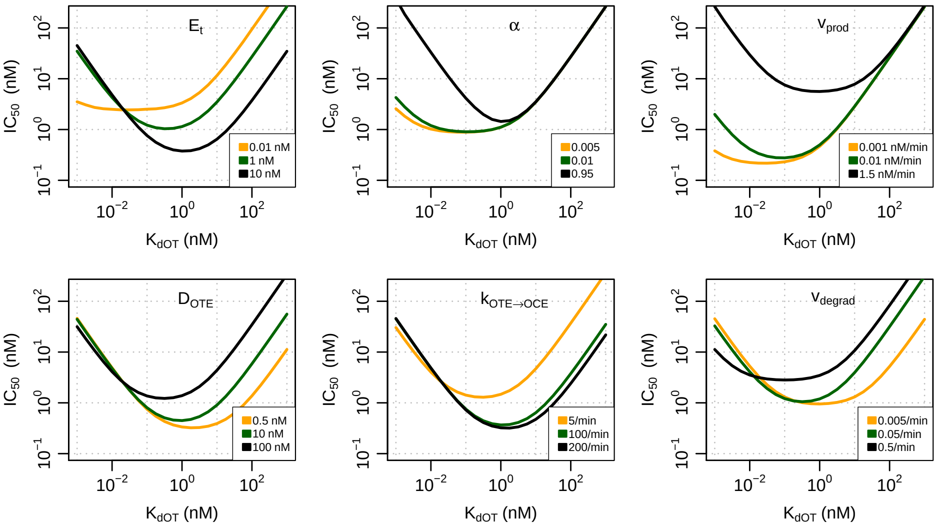
```
> IC50(KdOT=0.1,param=parms)

    IC50
1.218908
```

For a sequence of K_{dOT} values one can calculate the corresponding IC_{50} values and obtain Fig. 2c from the main manuscript, see vignette 2 (**Correct name?**). Figure S2 shows IC_{50} as a function of K_{dOT} for various parameter values. It can be seen that the optimum affinity, quantified by K_{dOT}, changes as parameters are changed. A larger value of K_{dOT} correponds to a better affinity for the oligonucleotide.

2 Deterministic simulation of the ASO model

Given an inital setting of concentrations for the eight variables, the Eqs. (1)-(8) can be solved in time by using the vode() function from the deSolve package (v1.10-4). The Eqs. (1)-(8) are written in the function diffASO(), which takes time, concentrations and parameters as input. Since the righthand-sides of Eqs. (1)-(8) are independent of time diffASO() returns the same



Supplementary Figure S2: The optimum affinity is dependent on the parameter settings. In the panels the IC_{50} concentration is plotted against the binding affinity quantified by D_{OT} for various parameters. From top-left to bottom-right we have varied the total RNAse H concentration (E_t), alpha, the rate of target cleavage ($k_{OTE \to OCE}$), the target production (v_{prod}) and degradation (v_{degrad}), and the dissociation constant for the OTE complex (D_{OTE}).

value for all timepoints. However, for use in vode() diffASO() needs to have time as input.

```
> init <- c(T=parms['vprod']/parms['vdegrad'],OT=0,OTE=0,E=parms['Et'],
+             O=100,OCE=0,OC=0)
> names(init) <- c('T','OT','OTE','E','O','OCE','OC')
> diffASO(1,init,parms)

[[1]]
[1] -100  100     0     0 -100     0     0

> diffASO(100,init,parms)

[[1]]
[1] -100  100     0     0 -100     0     0

> TimeSteps <- seq(0,3.5,by=0.05)/60 #time from 0 to 3.5 seconds
> solASO <- vode(init,TimeSteps,diffASO,parms)
> signif(solASO[1:10,],3)
```

```
          time    T     OT      OTE     E     O      OCE       OC
 [1,] 0.000000 5.00 0.0000 0.000000 1.000 100.0 0.00e+00 0.00e+00
 [2,] 0.000833 4.92 0.0825 0.000155 1.000  99.9 3.60e-07 3.47e-08
 [3,] 0.001670 4.84 0.1630 0.000562 0.999  99.8 2.33e-06 3.94e-07
 [4,] 0.002500 4.76 0.2420 0.001150 0.999  99.8 6.71e-06 1.67e-06
 [5,] 0.003330 4.68 0.3200 0.001870 0.998  99.7 1.38e-05 4.61e-06
 [6,] 0.004170 4.60 0.3960 0.002690 0.997  99.6 2.35e-05 9.98e-06
 [7,] 0.005000 4.53 0.4710 0.003560 0.996  99.5 3.57e-05 1.85e-05
 [8,] 0.005830 4.45 0.5440 0.004470 0.995  99.5 5.00e-05 3.10e-05
 [9,] 0.006670 4.38 0.6170 0.005410 0.995  99.4 6.60e-05 4.78e-05
[10,] 0.007500 4.31 0.6870 0.006350 0.994  99.3 8.34e-05 6.95e-05
```

The time is measured in minutes and concentrations in nM. From a time-serie as the one above Fig. 2a can be made, see vignette 2 **(Correct name??)**.

3 Stochastic simulation of the ASO model

The stochastic simulation of the ASOmodel is carried out by use of the ssa() function from the GillespieSSA package (v.0.5-4). The inputs of ssa are an initial state vector (x0), which is the initial number of molecules, a propensity vector (a), which denotes the different states of the system, a state-change matrix (nu), which is the change in number of molecule (rows) if a reaction occur (column), the model-parameters (parms) and the final time (tf).

```
> library(GillespieSSA)
> parms1 <- c(k1 = 2E-5,k2 =50E-5 ,vt = 150,  kd = 0.04,
+                 kE = 2, km1 =0.06, km2=2, k3 = 0.1)
> x0 <- c(Tt=parms1["vt"]/parms1["kd"],
+         OT=0,OTE=0,E=1e3,O=1e5,OCE=0,OC=0)
> names(x0) <- c('Tt','OT','OTE','E','O','OCE','OC')
> a <-  c("vt","k1*O*Tt","kd*Tt","km1*OT","km2*OTE","kd*OT",
+          "k2*OT*E","kd*OTE","kE*OTE","k3*OC","km2*OCE" )
> nu <- matrix(0,7,length(a))
> dimnames(nu) <- list(names(x0),a)
> #T
> nu['Tt',c('vt','km1*OT')] <- 1
> nu['Tt',c('k1*O*Tt','kd*Tt')] <- -1
> #OT
> nu['OT',c('k1*O*Tt','km2*OTE')] <- 1
> nu['OT',c('km1*OT','k2*OT*E','kd*OT')] <- -1
> #OTE
> nu['OTE',c('k2*OT*E')] <- 1
> nu['OTE',c('km2*OTE','kd*OTE','kE*OTE')] <- -1
> #E
> nu['E',c('km2*OTE','kd*OTE','km2*OCE')] <- 1
> nu['E',c('k2*OT*E')] <- -1
> #O
> nu['O',c('km1*OT','kd*OTE','kd*OTE','kd*OT','k3*OC')] <- 1
> nu['O',c('k1*O*Tt')] <- -1
> #OCE
> nu['OCE',c('kE*OTE')] <- 1
> nu['OCE',c('km2*OCE')] <- -1
> #OC
> nu['OC',c('km2*OCE')] <- 1
> nu['OC',c('k3*OC')] <- -1
> Gillespie <- ssa( x0=x0,# initial state vector
+       a=a, # propensity vector
+       nu=nu, # state-change matrix
+       parms = parms1, # model parameters
+       tf=1E3, # final time
+       method = "ETL" # SSA method
+ )
> # check that [O]+[OT]+[OTE]+[OCE]+[OC] = Ot at all times
> range(rowSums(Gillespie$data[,c('O','OT','OTE','OCE','OC')])-x0['O'])
```

```
[1] 0 0
```

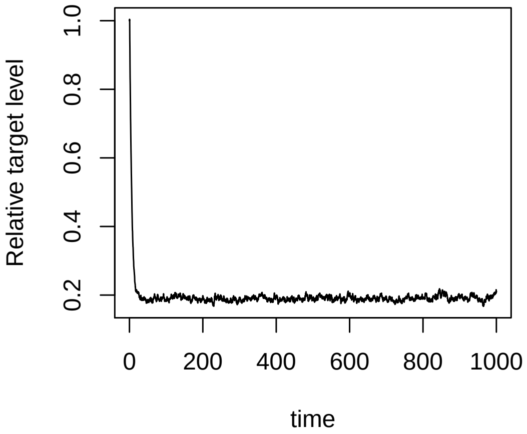
```
> # check that [E]+[OTE]+[OCE] = Et at all times
> range(rowSums(Gillespie$data[,c('OTE','OCE','E')])-x0['E'])
```

```
[1] 0 0
```

```
> # plot of the time-trase for Trel
> plot(Gillespie$data[,1],rowSums(Gillespie$data[,2:4])/(parms1["vt"]/parms1["kd"]),
+                 xlab='time',ylab='Relative target level',type='l')
```

After a while the relative target concentration reaches a plateau. The mean of T_{rel} within the plateu can be calculated through the function Trel():

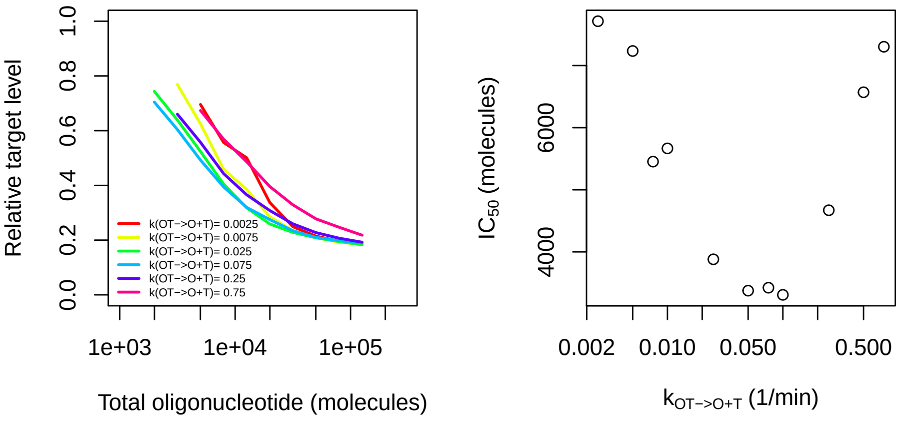
```
> Trel <- function(Ot,km1=0.06){
+         parms1['km1'] <- km1; parms1['k3'] <- km1/0.6
+         x0['O'] <- Ot
+         Gillespie <- ssa(x0=x0,a=a,nu=nu,parms = parms1,tf=2E2,method = "ETL")
+         data <- rowSums(Gillespie$data[,2:4])/x0['Tt']
+         Tmean <- mean(data[200:nrow(Gillespie$data)])
+         Tsd <- sd(data[200:nrow(Gillespie$data)])
+         return(list(Trel=Gillespie$data,Tstat=c('O'=x0['O'],TrelM=Tmean,TrelSD=Ts
+ }
```



Supplementary Figure S3: Bla

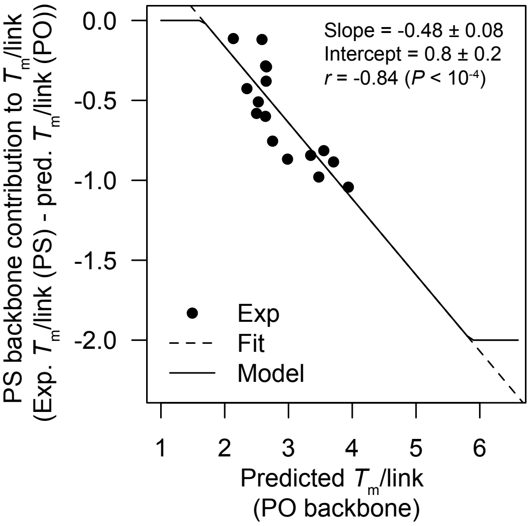
Using Trel() we can generate dose-response curves, and frome these the IC_{50}-values can be calculated using the IC50stoc() function.

```
> lseq <- c(1,2.5,5,7.5)
> KM <- c(1E-3*lseq[-1],1E-2*lseq,1E-1*lseq)
> #### Generation of dose-response curves
> DRcurve <- lapply(KM,function(kmi){
+                                  sapply(10^seq(2.5,6,by=0.2),function(i) Trel(i,kmi
> DRc <- lapply(DRcurve,function(x) x[,!is.na(x[3,])] )
> #### Calculation of IC50
> IC50_KM <- sapply(1:length(DRc),function(x){IC50stoc(DRc[[x]][2,],DRc[[x]][1,])})
> par(mfcol=c(1,2))
> N <- length(DRc)
> plot(DRc[[1]][1,],DRc[[1]][2,],ylab='Relative target level',ylim=c(0,1),
+                 xlab='Total oligonucleotide (molecules)',log='x',type='l',
+                 col=rainbow(N)[1],xlim=c(1E3,3E5),lwd=2)
> legend('bottomleft',paste('k(OT->O+T)=',signif(KM[seq(1,11,by=2)],2)),
+                 col=rainbow(N)[seq(1,11,by=2)],lwd=2,bty='n',cex=0.5)
> for(i in seq(3,11,by=2)){
+         points(DRc[[i]][1,],DRc[[i]][2,],col=rainbow(N)[i],type='l',lwd=2)
+ }
> plot(KM,IC50_KM,log='x',ylab=expression(IC[50]~'(molecules)'),
+                 xlab=expression(k['OT->O+T']~'(1/min)'))
```

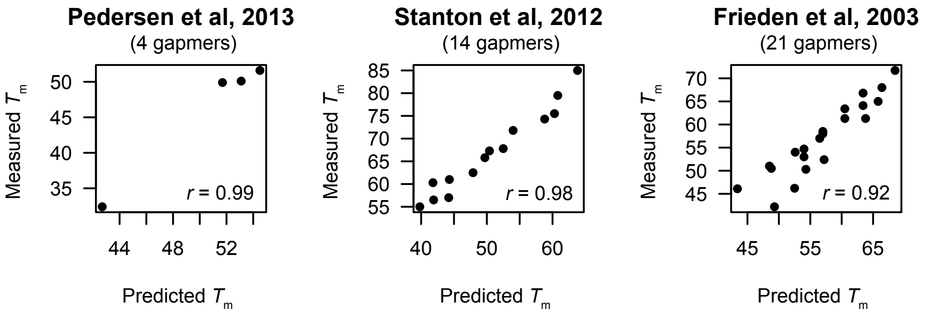


Supplementary Figure S4: Bla

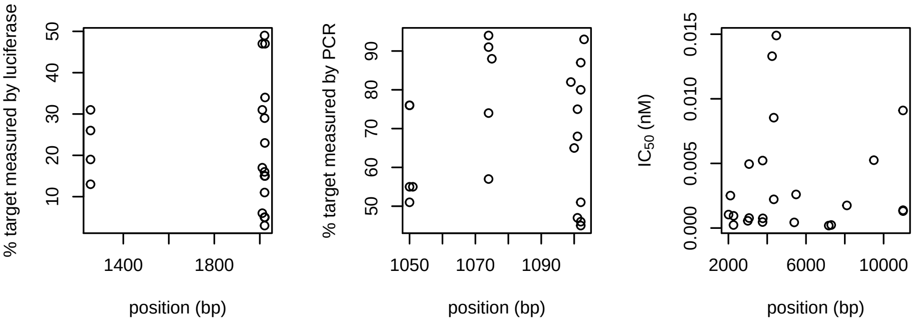
4 Prediction of the melting temperatur (T_m) of the OT complex



Supplementary Figure S5: Bla bla



Supplementary Figure S6: Measured melting temperature T_m vs predicted melting temperatur. There are a clear correlations (r $>$ 0.92, P $<$ 0.01, Pearson's correlation) between predicted and measured T_m. Pedersen et al: 4 LNA-modified oligonucleotides targeting apolipoprotein B (this work), Stanton et al: 14 LNA-modified oligonucleotides targeting the glucocorticoid receptor (Stanton et al., 2012). Frieden et al: 21 LNA-modified oligonucleotides targeting the luciferase firefly gene (Frieden et al., 2003).



Supplementary Figure S7: For each oligonucleotide the percent-target (left and middle) or the IC_{50} value (right) is plotted as a function of oligonucleotide binding-position in the target.