

Figure 7:

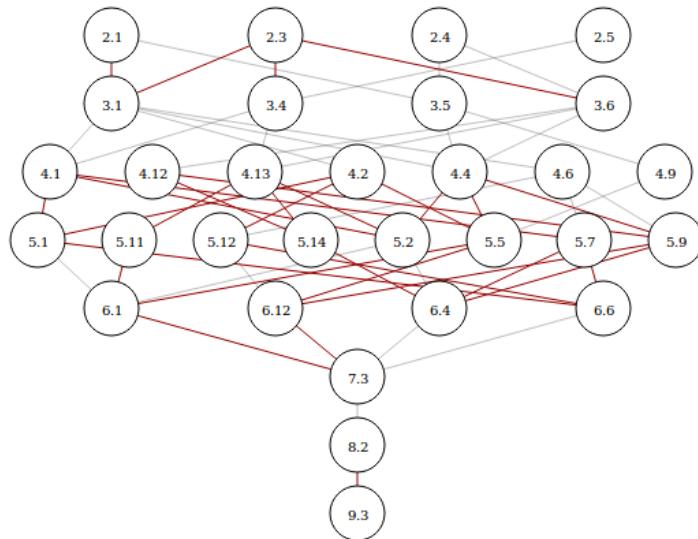


Figure 8:

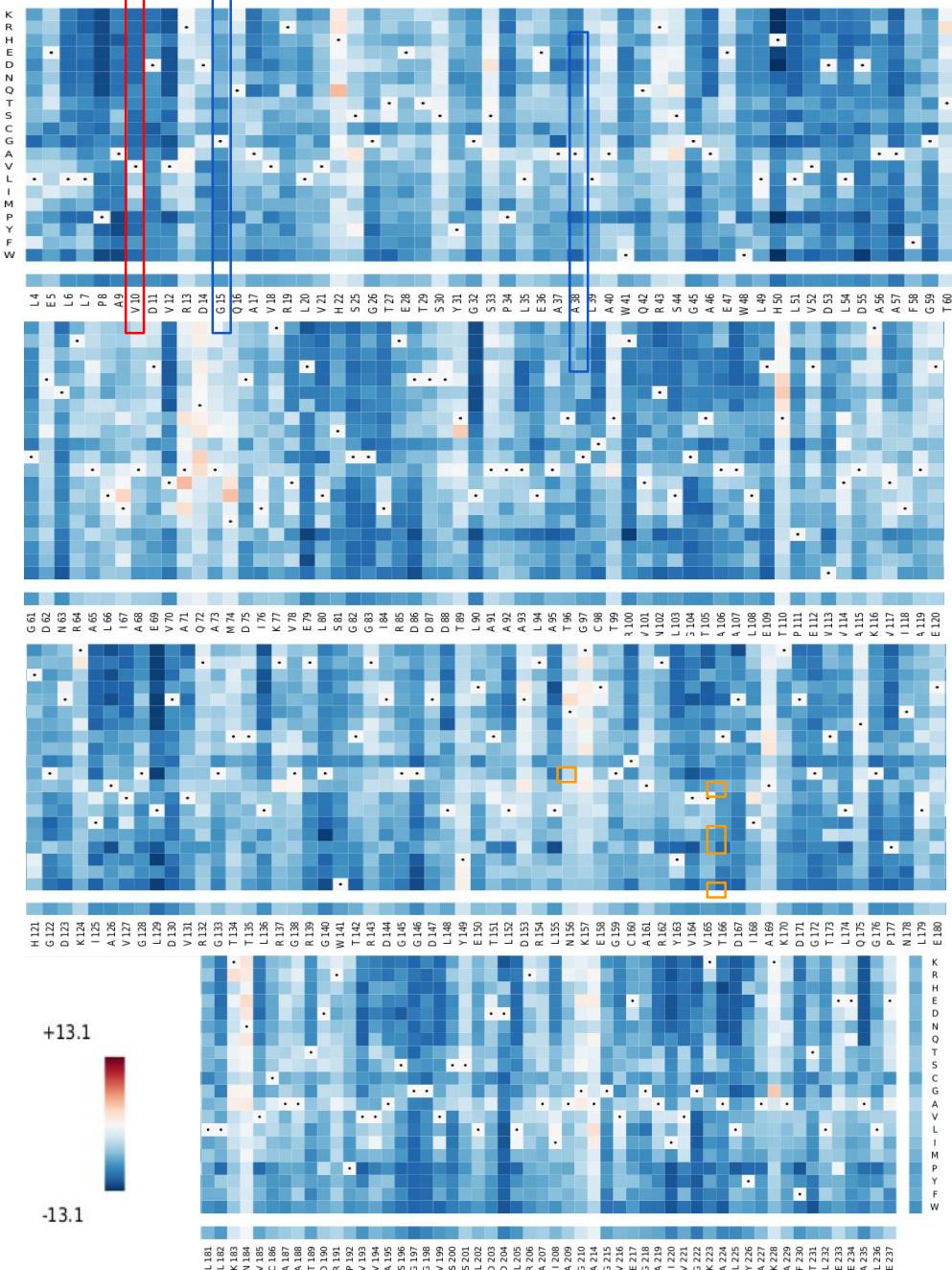


Figure 9:

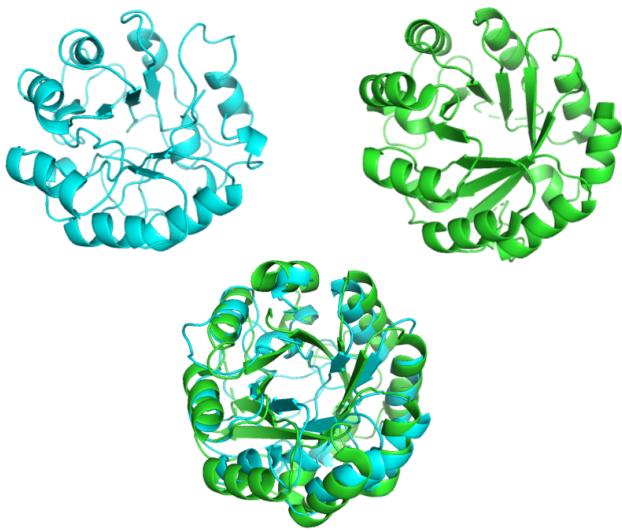


Figure 10:

subHisA

Corynebacterium diphtheriae *Actinomyces car* (PDB:4X2R)

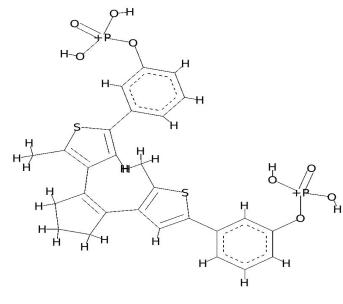
subTrpF

Arthrobacter aurescens (PDB:4WD0)

PriB

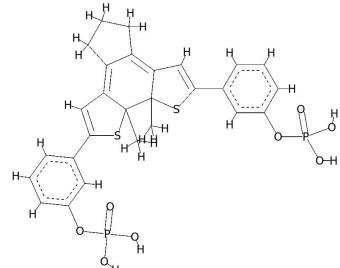
Streptomyces ipomoeae, *Streptomyces sviceus* (PDB:4U28,4TX9)

TrpF controls *Jonesia denitrificans* (PDB:4WUI) *Chlamidya trachomatis*, *Streptomyces sp. Mg1* TrpF and *Actinomyces odontolyticus* were included



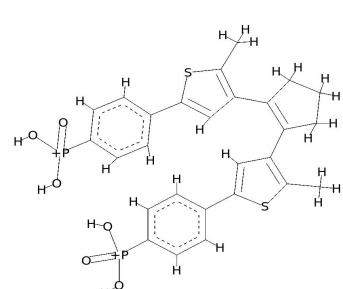
13

DTE-meta-phosphate(dte6_Open form)



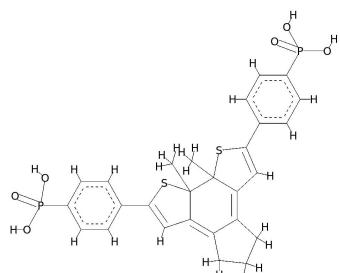
14

DTE-meta-phosphate(dte6_Closed form)



15

DTE-Para-Phosphonate(dte13_Closed form)



16

DTE-para-phosphonate(dte13_Closed form)

Figure 11: Substatos químicamente similares a los sustratos de PriA (parte 1)

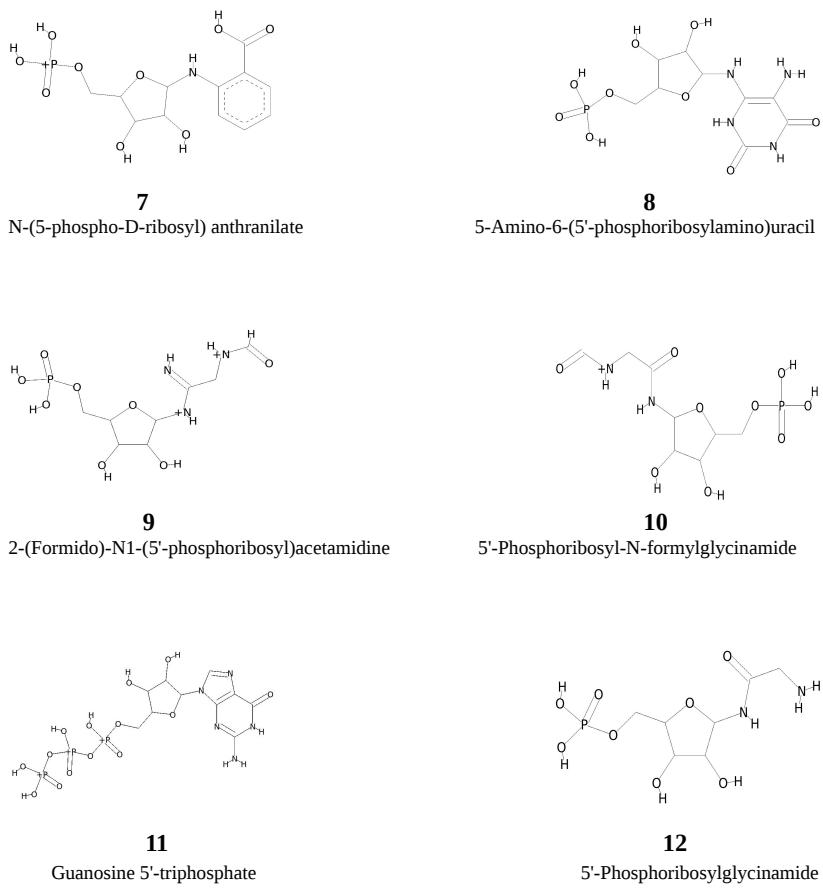
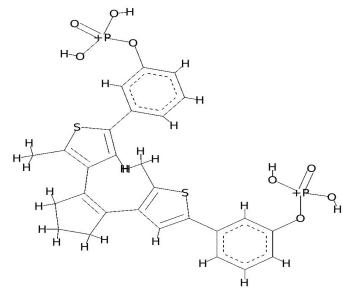
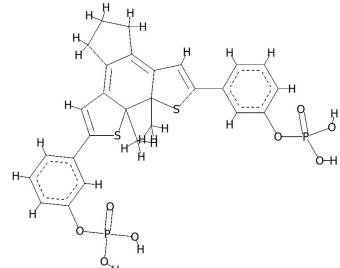


Figure 12: Substatos químicamente similares a los sustratos de PriA (parte 2)



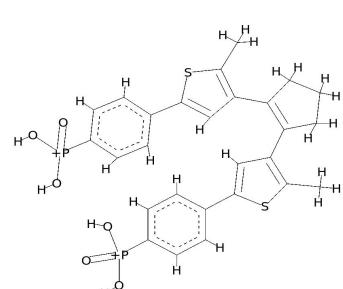
13

DTE-meta-phosphate(dte6_Open form)



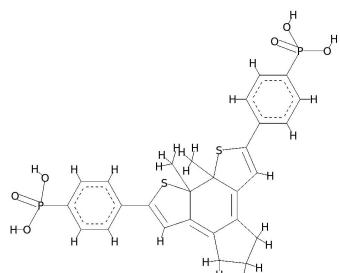
14

DTE-meta-phosphate(dte6_Closed form)



15

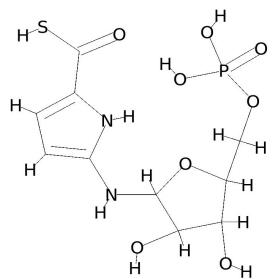
DTE-Para-Phosphonate(dte13_Closed form)



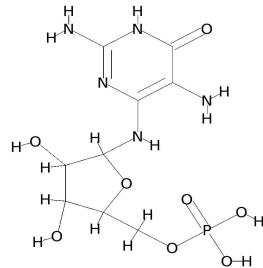
16

DTE-para-phosphonate(dte13_Closed form)

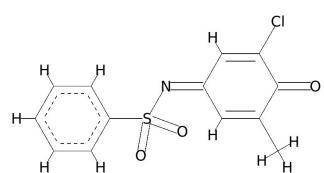
Figure 13: Substatos químicamente similares a los sustratos de PriA (parte 3)



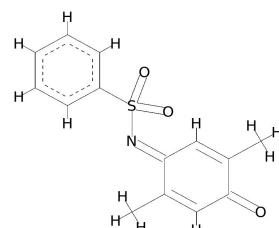
17
4N'-(5'-phosphoribosyl) 4-aminopyrrole
-2-carboxilate



18
2,5-di-amino-6-ribosylamino-4
(3H)-pyrimidinone 5'-phosphate



19
(E)-N-(3-chloro-5-methyl-4-oxocyclohexa-
2,5-dienylidene)benzenesulfonamide



20
2,5 dimethyl-N-(4-oxocyclohexa-
2,5-dienylidene)benzenesulfonamide

Figure 14: Substatos químicamente similares a los sustratos de PriA (parte 4)

Sustratos similares a PRA y PROFAR sugeridos por distancias de Tanimoto o probados previamente

S1, S2, ..., S20 sustratos fueron recolectados de la literatura y las predicciones de la quimioinformática. S3 PRA y S7 PROFAR son sustratos nativos, S13-S16 son sustratos activados por la luz, S17 PRAP, S18 Compuesto V, se encontraron en la literatura, S6 GMP, S11 GTP y otros fueron sugeridos por chemoinformatics.

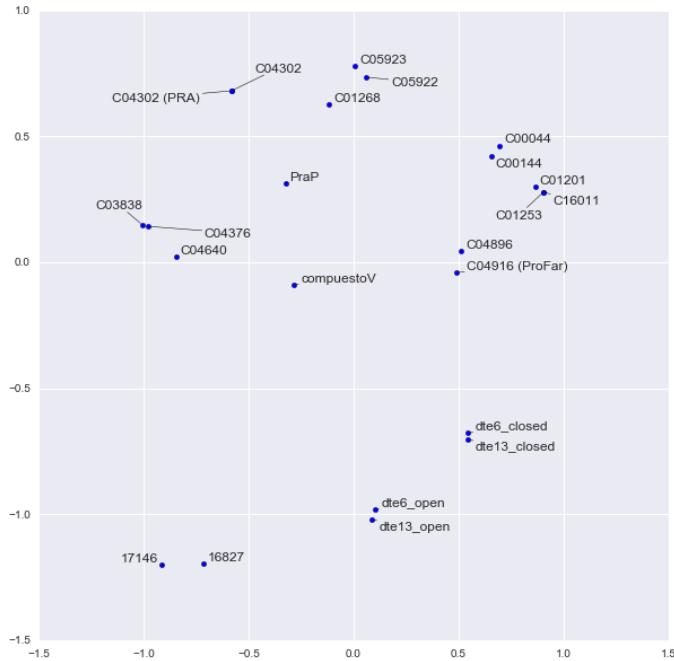


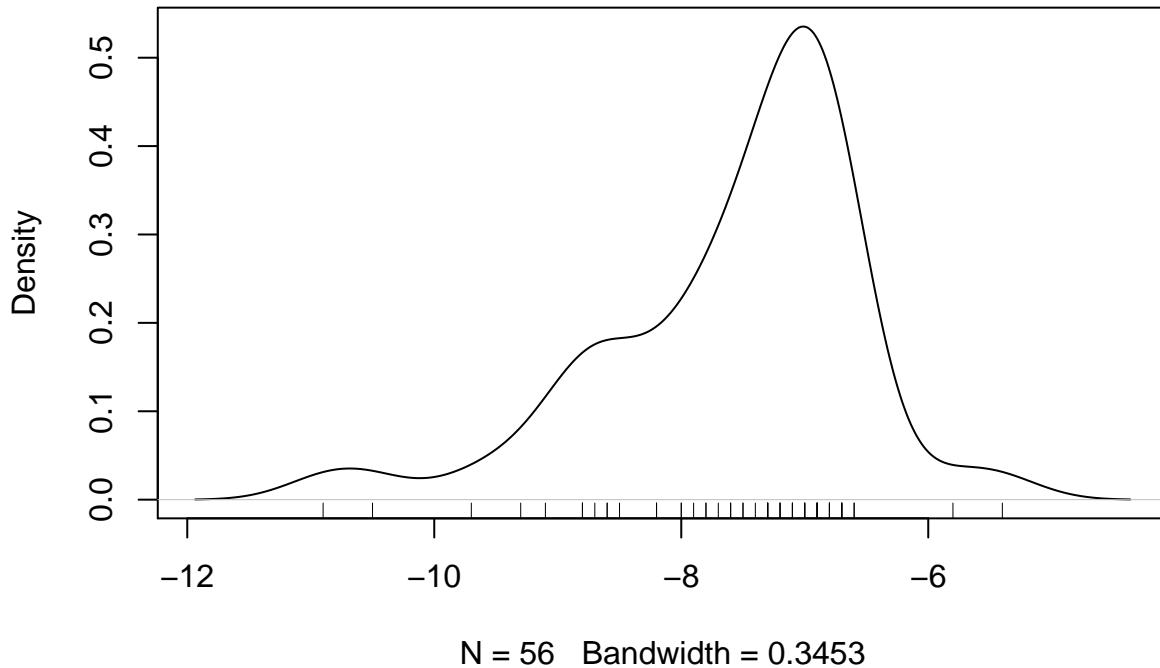
Figure 15:

On next figure we can see their chemical structures.

```
### Docking between PriA enzymes and selected substrates  
Para diversas enzimas PriA de Streptomyces se realizaron simulaciones de docking. Se incluyeron también como controles enzimas TrpF provenientes de Streptomyces Mg1, Jonesia denitrificans. Los procedimientos pueden ser consultados en Docking Protocols
```

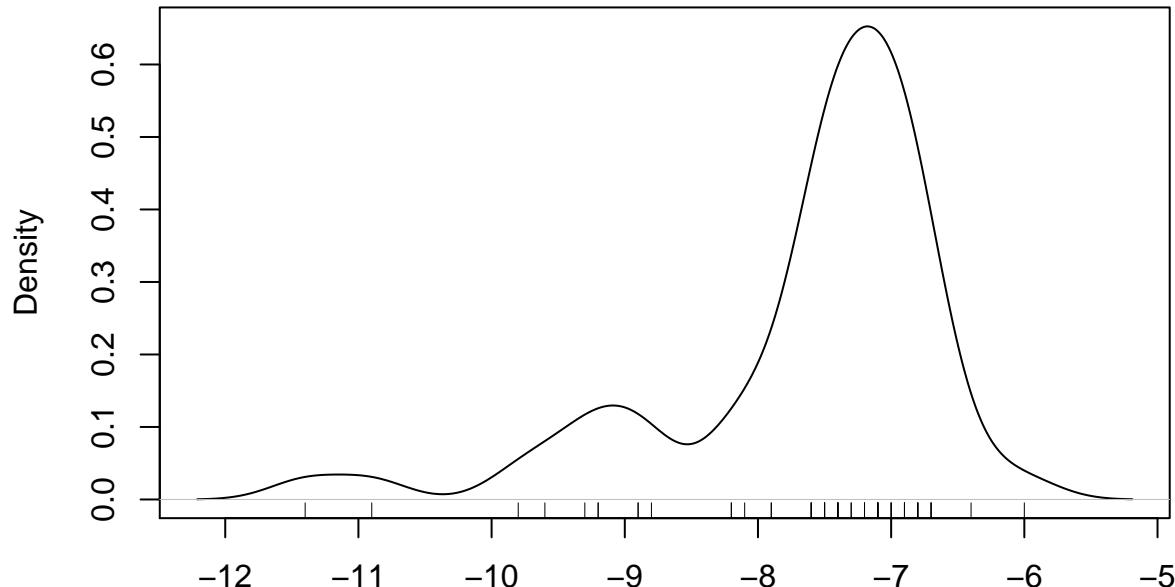
```
## Called from: eval(expr, envir, enclos) ## debug en <text>#4: plot(density(docking[,  
i], na.rm = T))
```

```
density.default(x = docking[, i], na.rm = T)
```



```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en  
<text>#4: plot(density(docking[, i], na.rm = T))
```

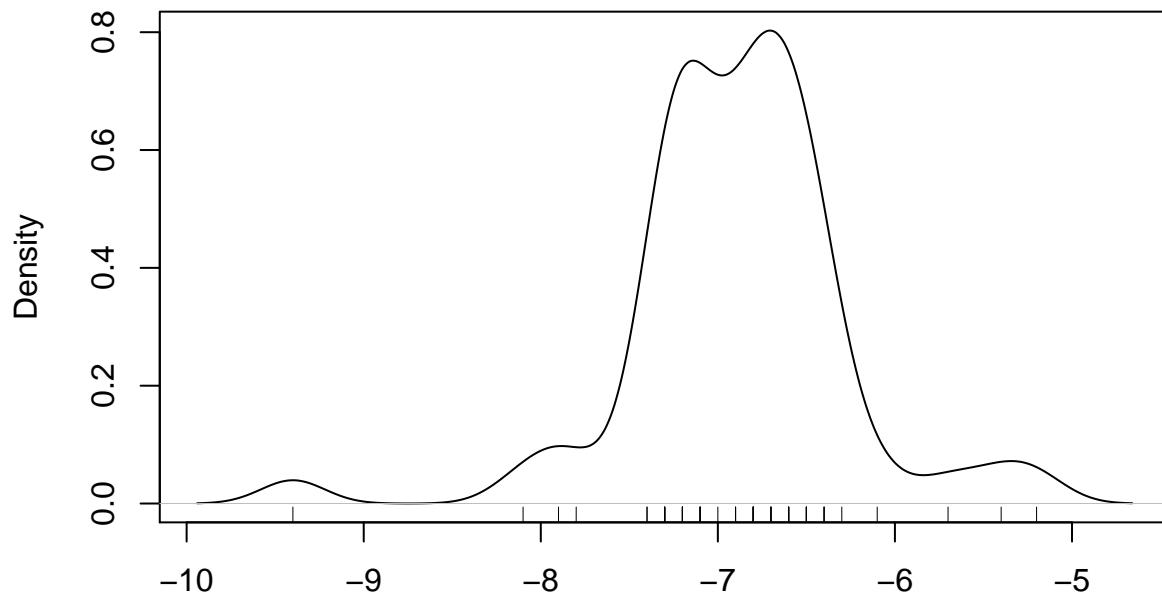
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.2702

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

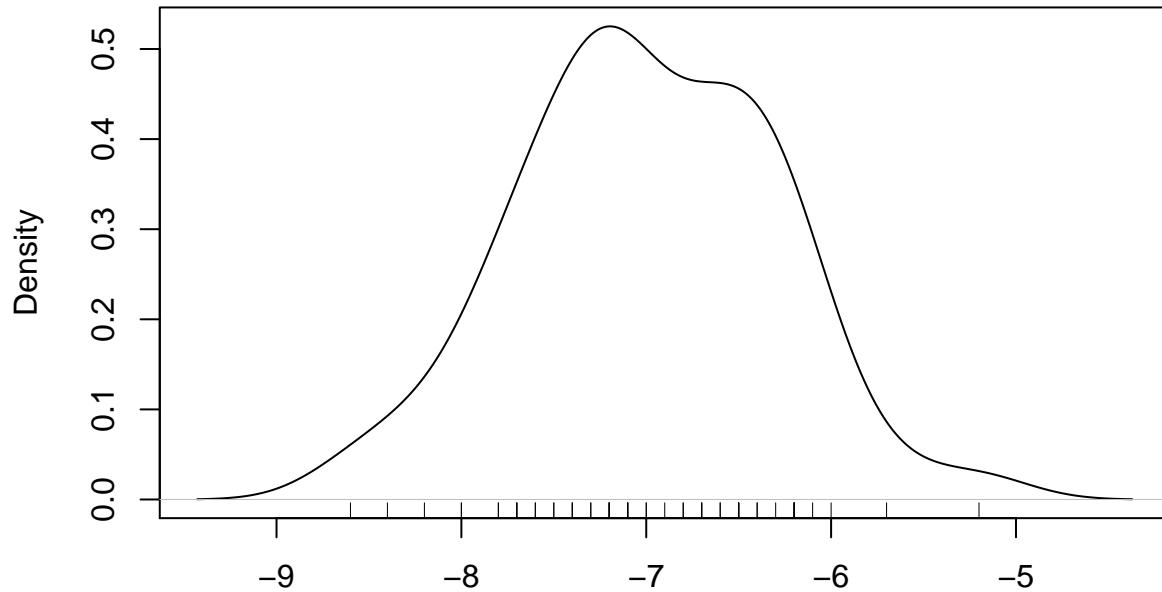
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.1802

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

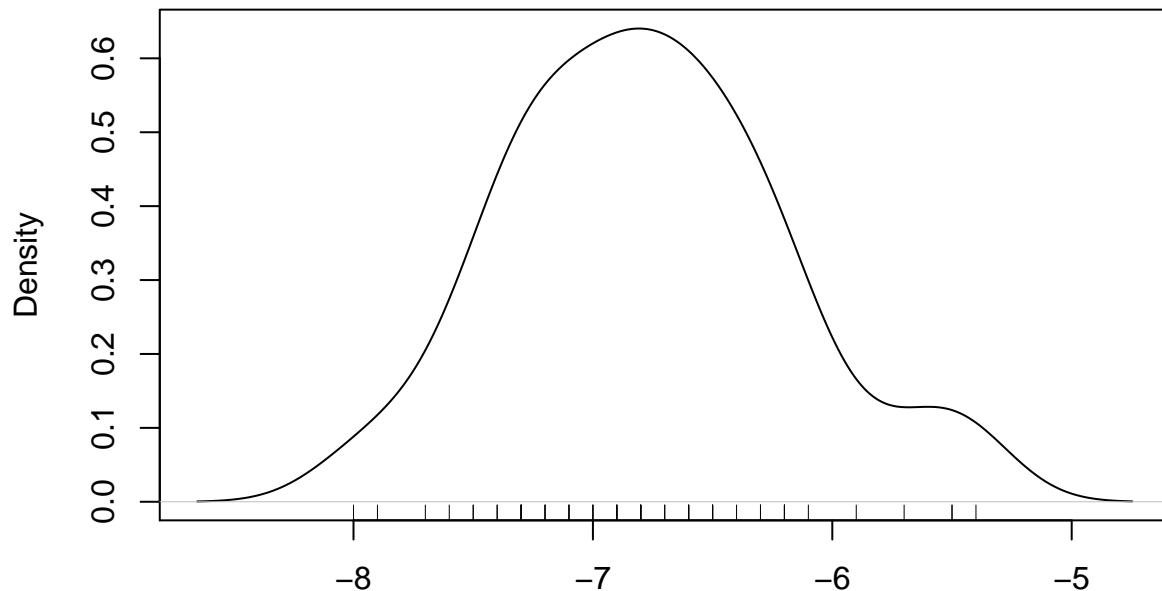
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.2764

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

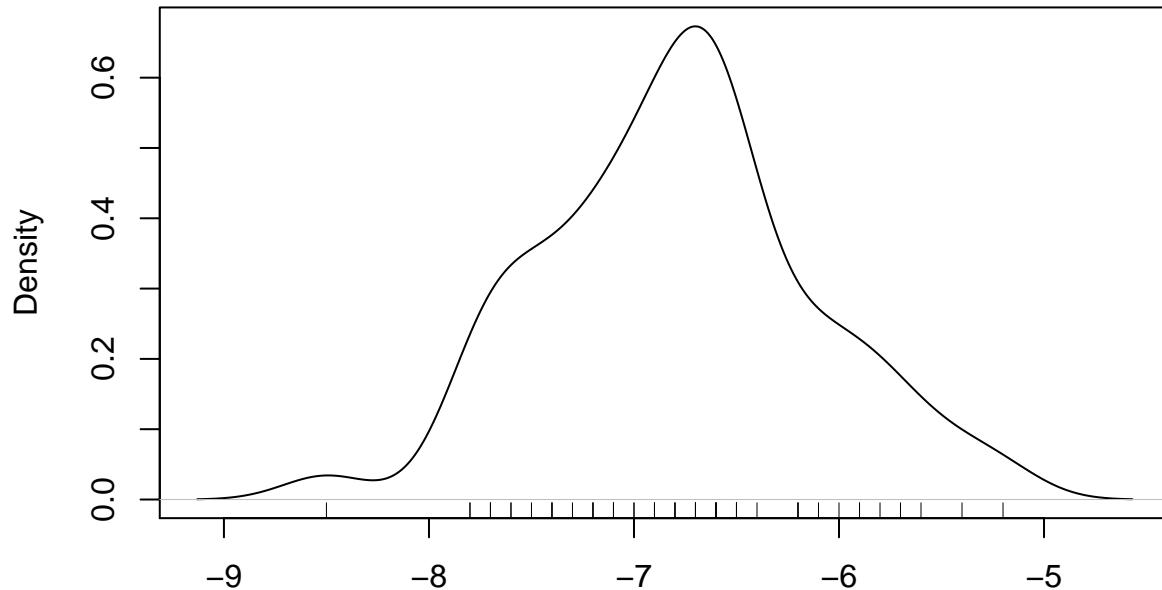
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.2177

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

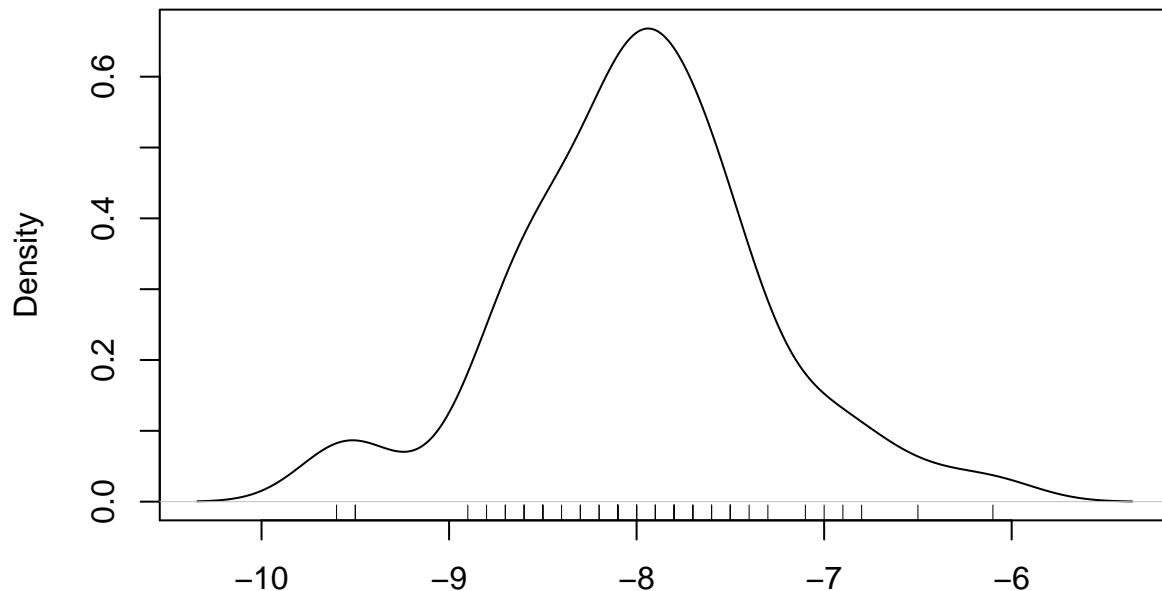
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.2102

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

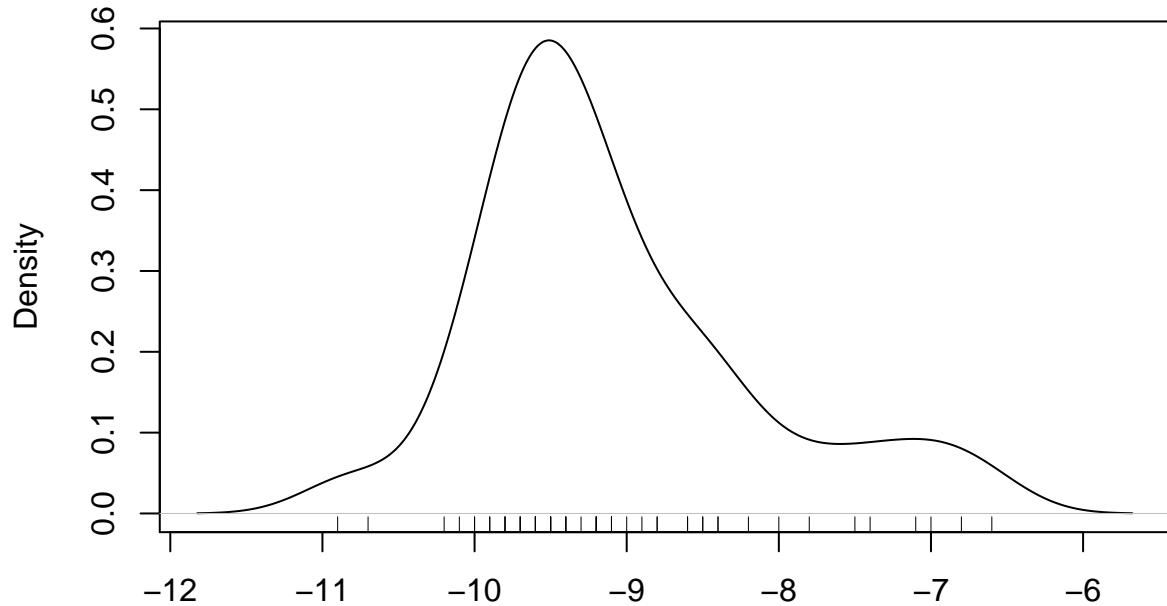
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.2477

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

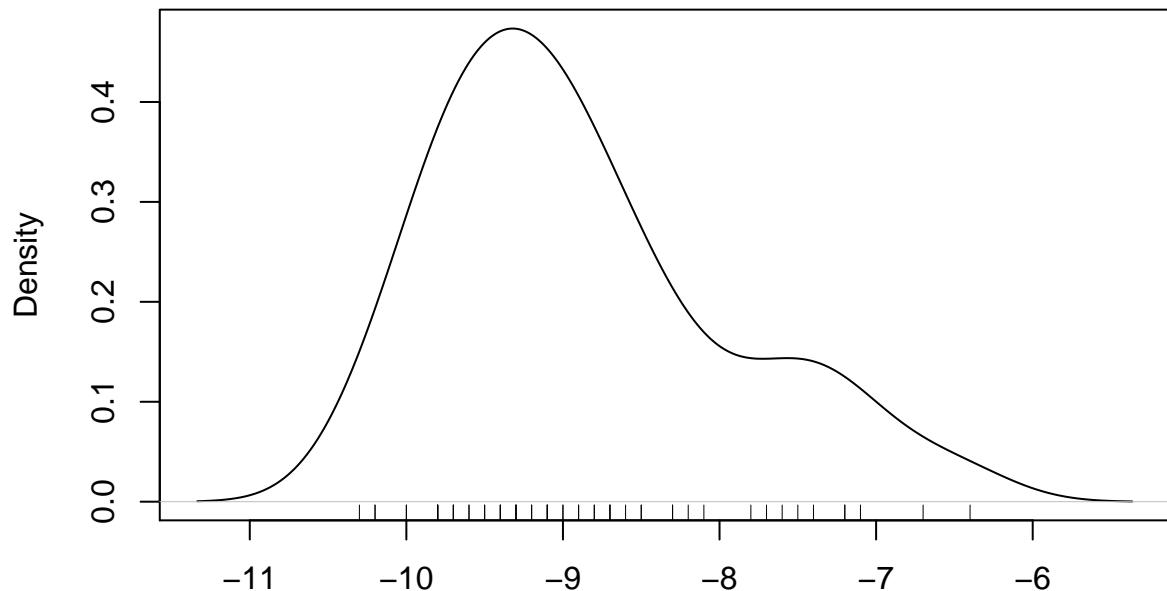
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.3078

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

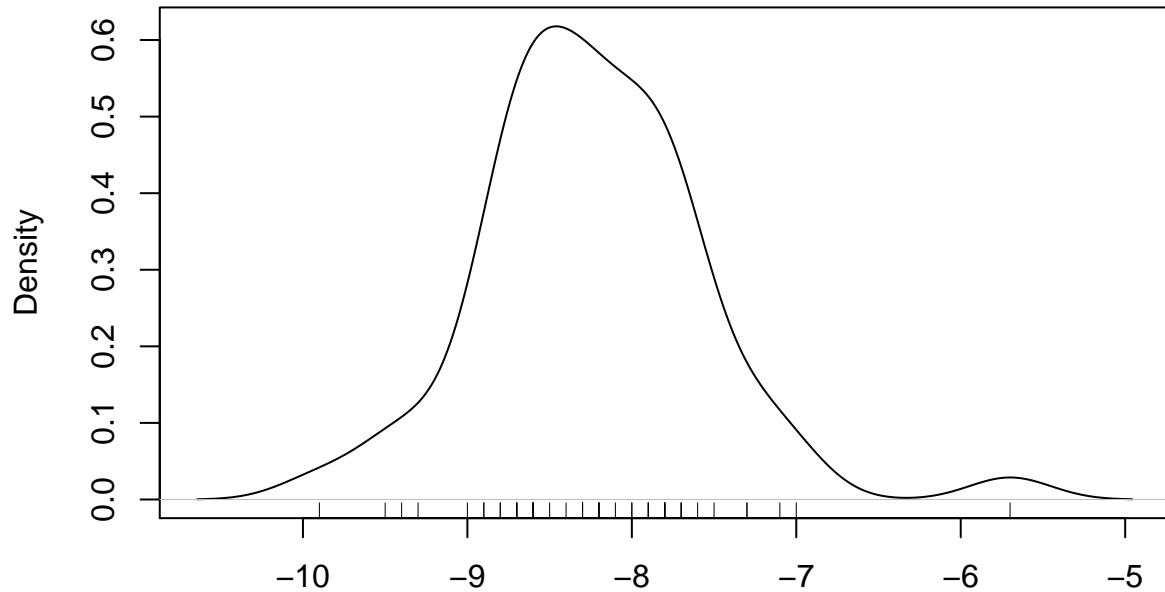
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.3453

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

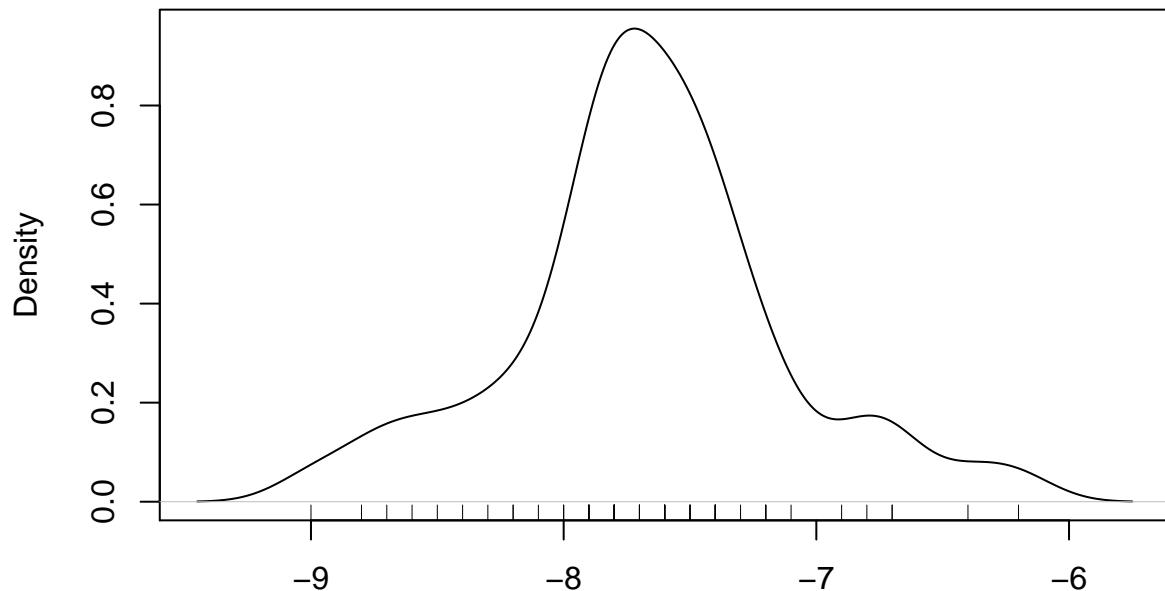
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.2477

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

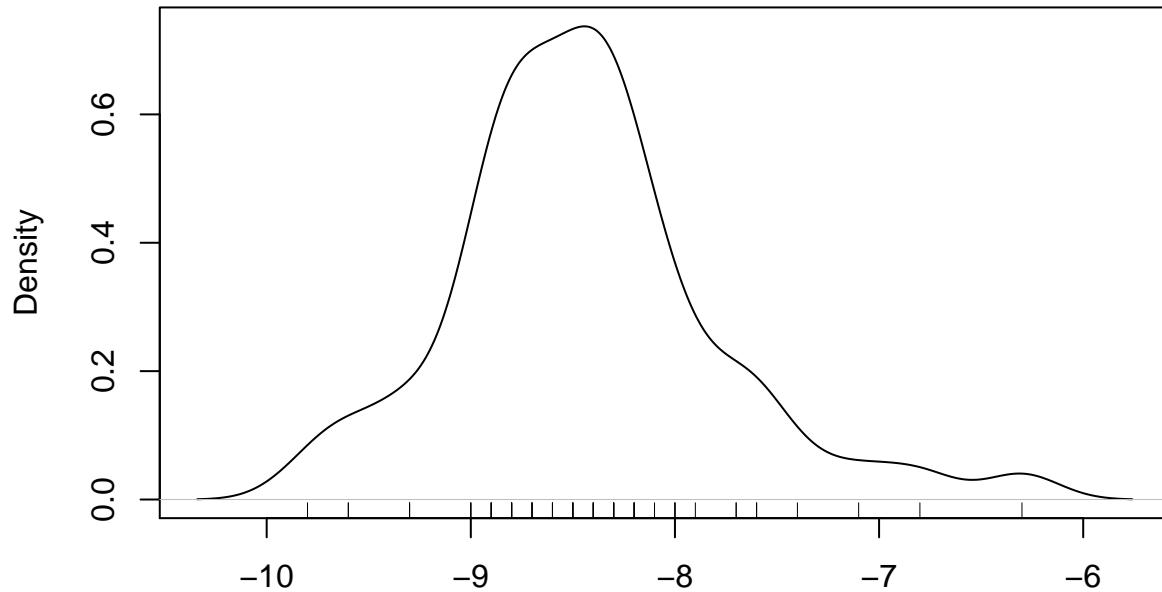
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.1501

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

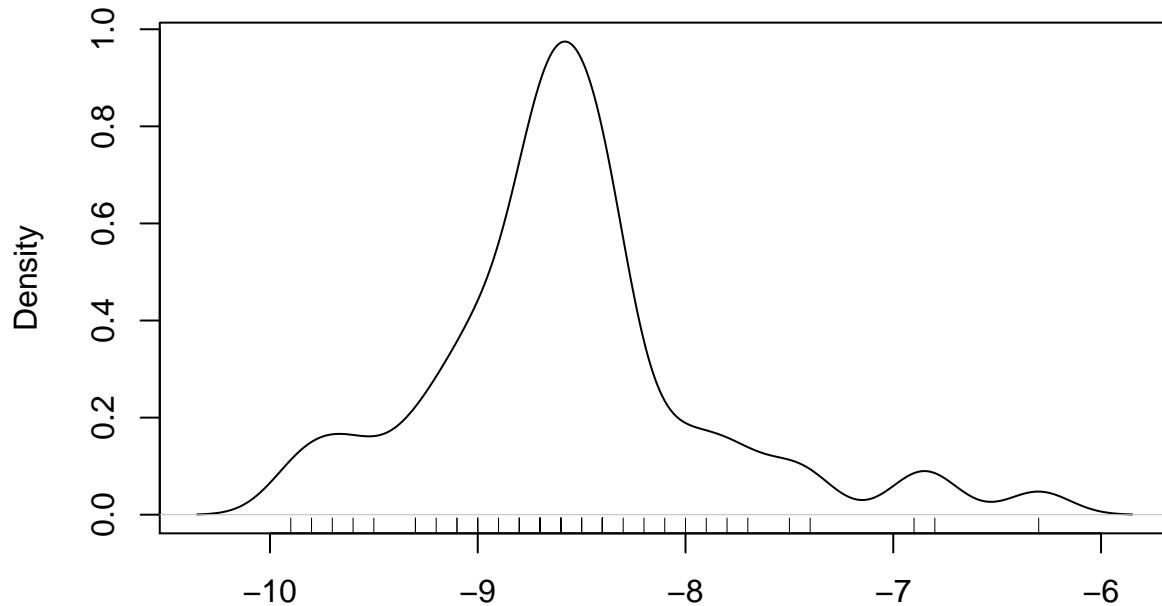
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.1802

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

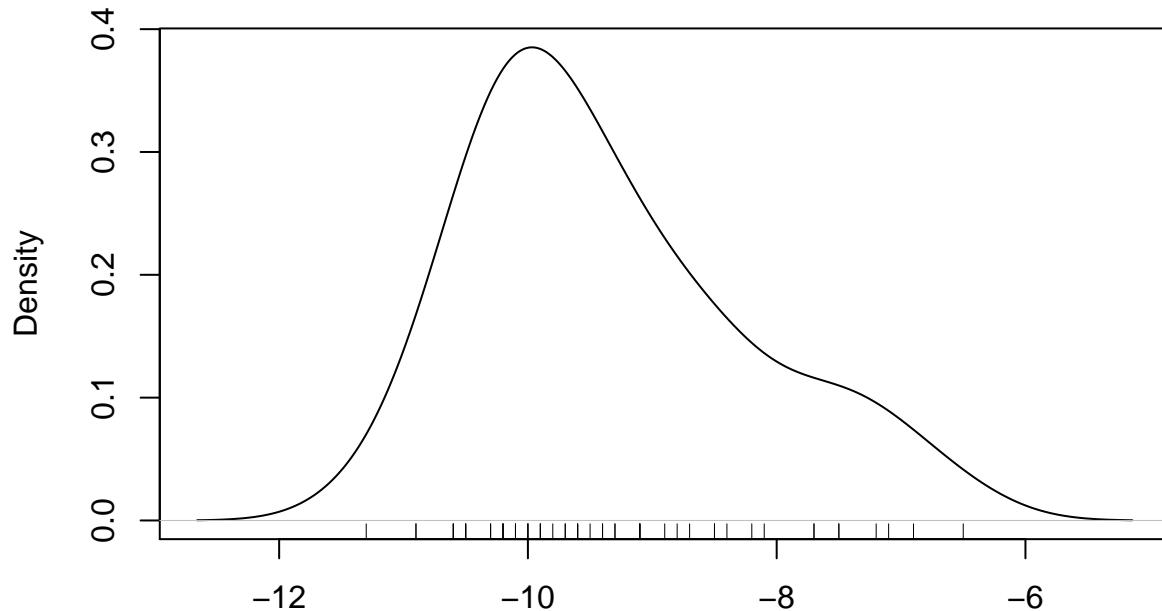
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.1501

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

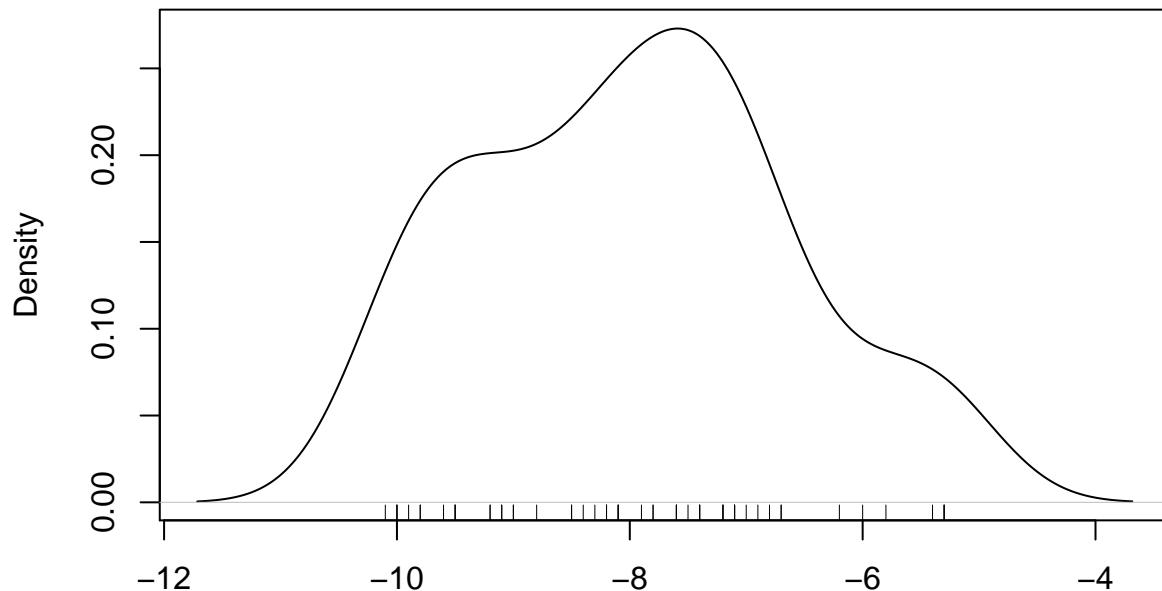
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.453

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

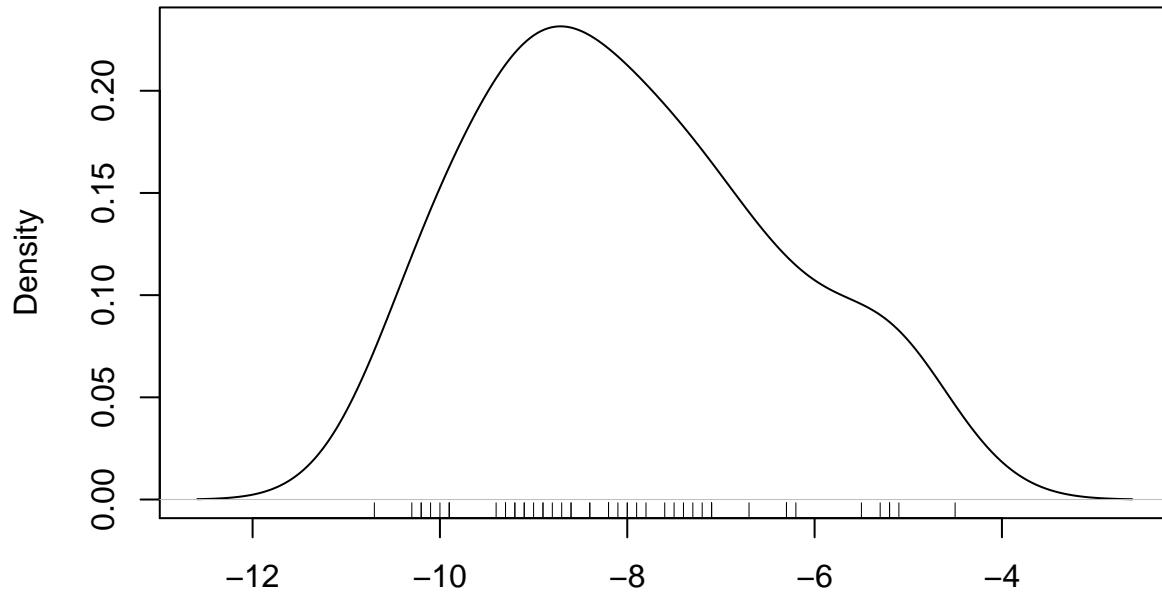
density.default(x = docking[, i], na.rm = T)



N = 56 Bandwidth = 0.5386

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

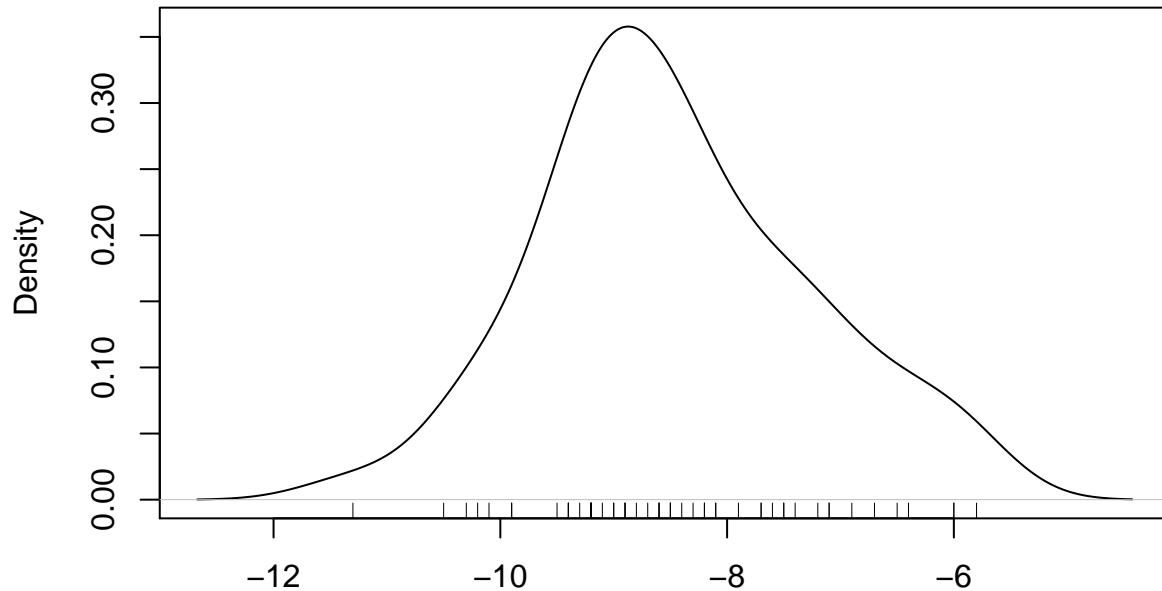
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.6305

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

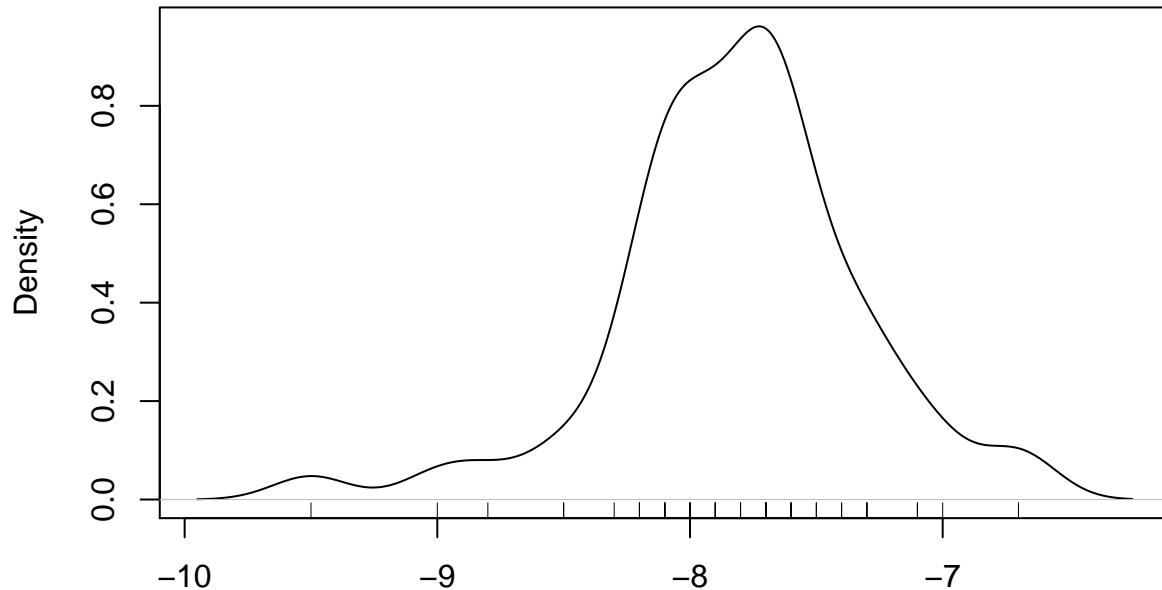
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.4579

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

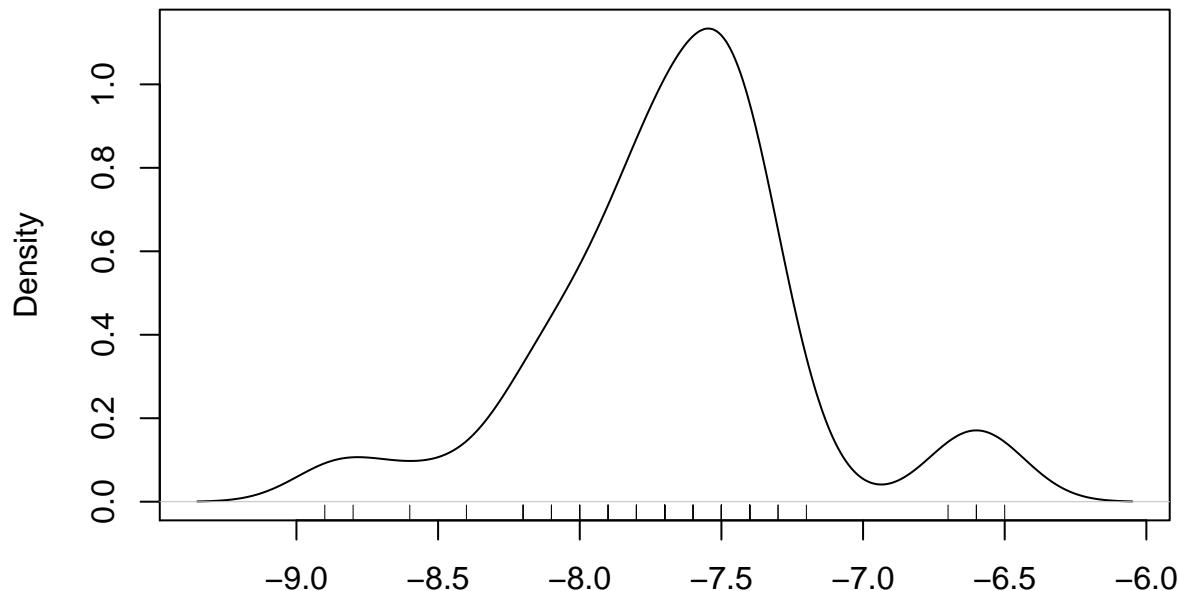
```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.1501

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser() ## debug en
<text>#4: plot(density(docking[, i], na.rm = T))
```

```
density.default(x = docking[, i], na.rm = T)
```



N = 56 Bandwidth = 0.1501

```
## debug en <text>#4: rug(docking[, i]) ## debug en <text>#4: browser()
r plot(cor(docking[-c(5,12,13,38,51),-1]))
```

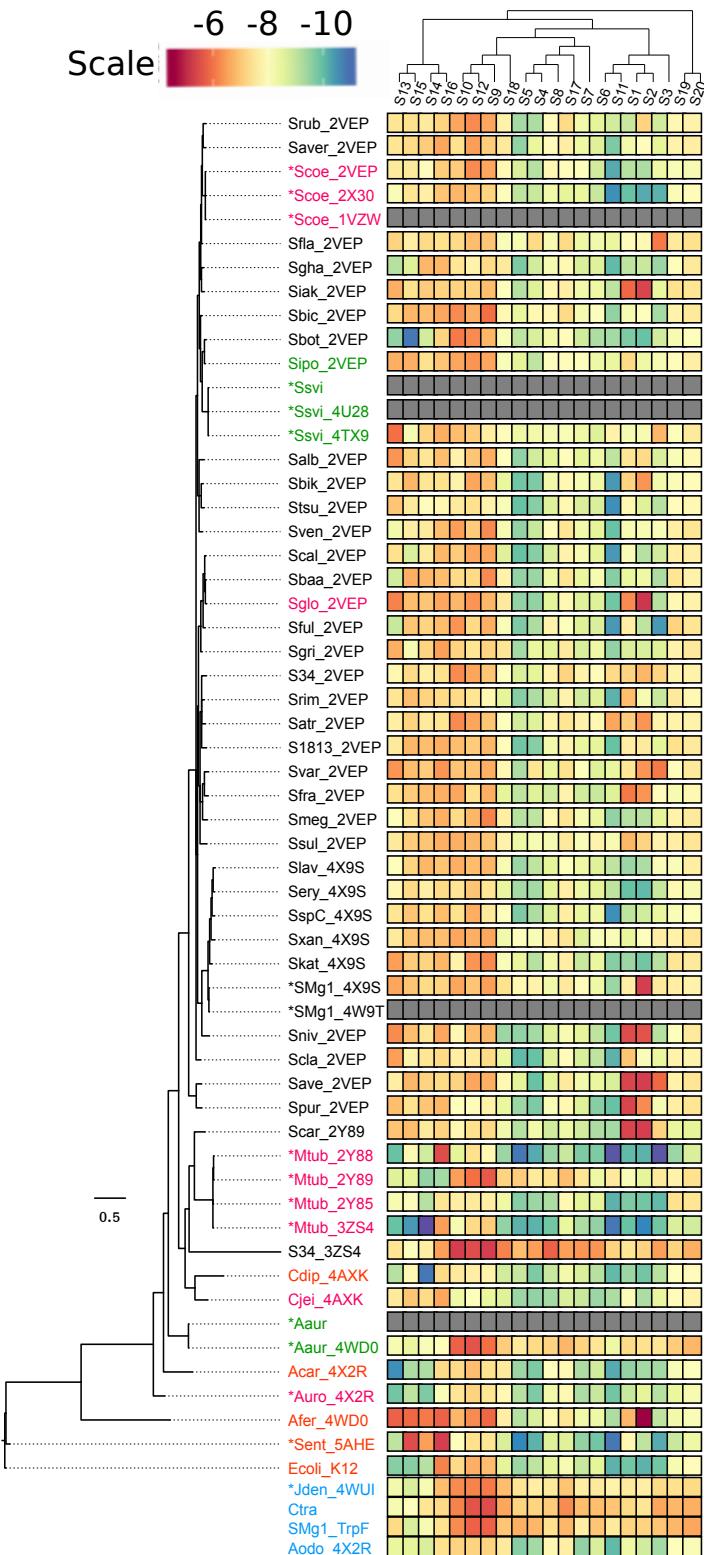



Figure 16:
28

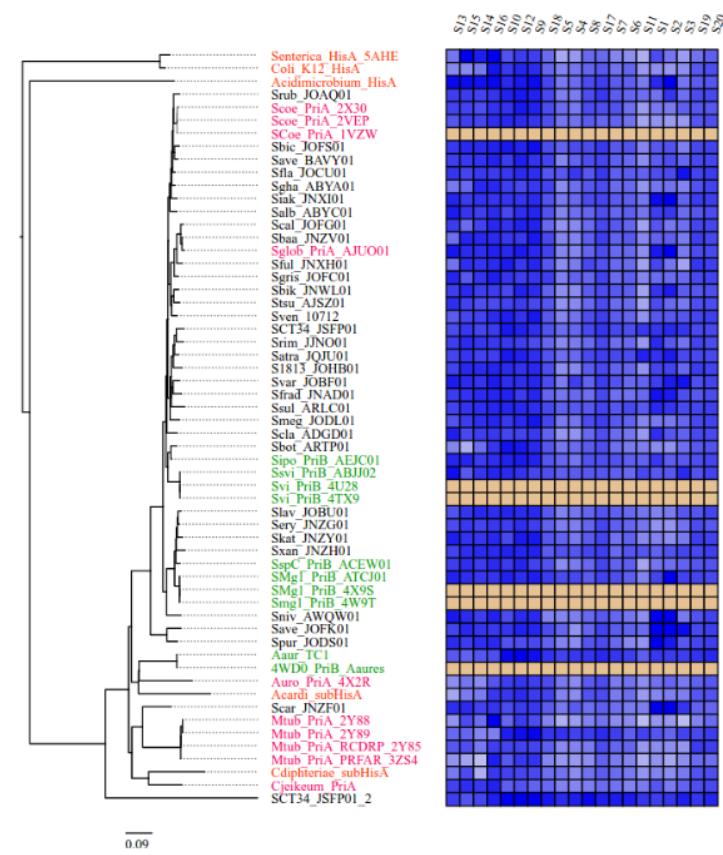
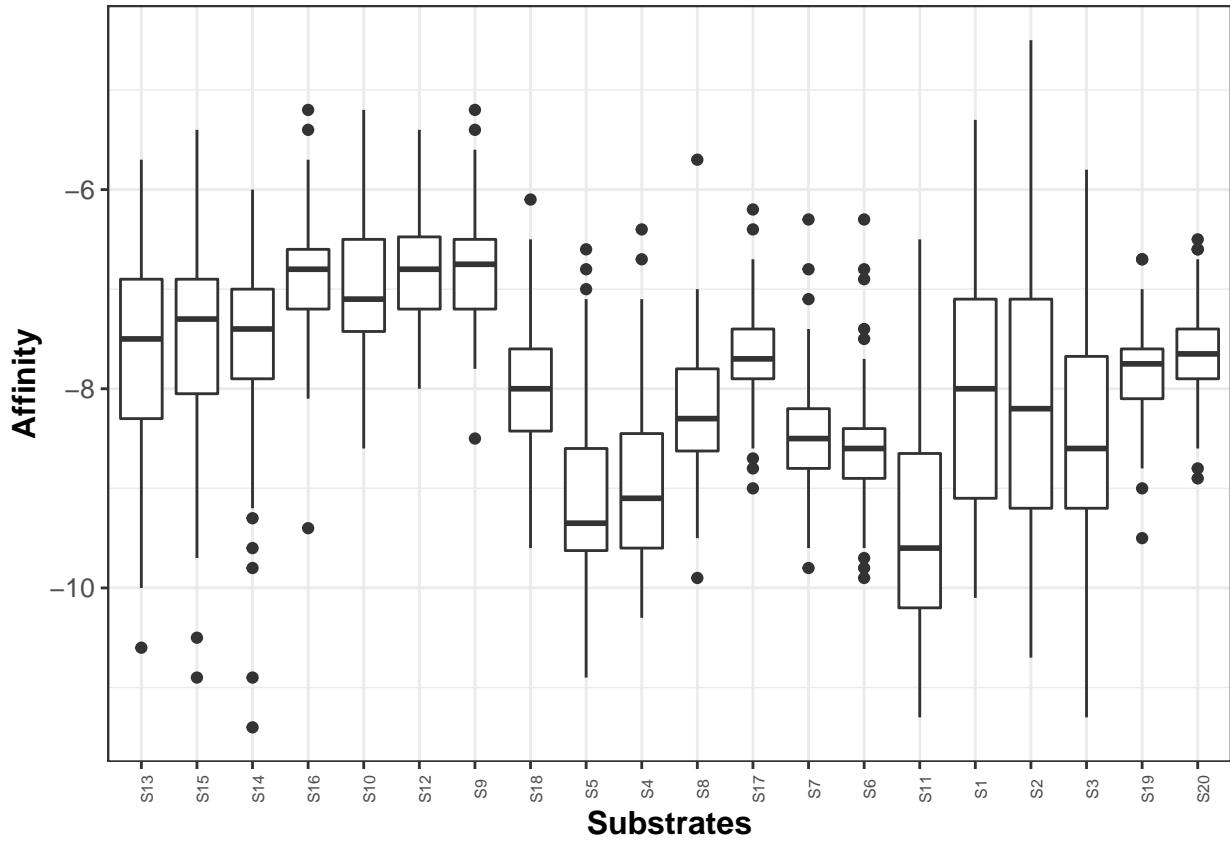


Figure 17:

GTP es el sustrato por el que PriA muestra mayor afinidad

```
## boxplot de los sustratos
ggplot(docking.m, aes(x=variable, y=value)) + labs(x = "Substrates", y = "Affinity",text = element_text)
## Warning: Removed 100 rows containing non-finite values (stat_boxplot).
```




```
#sessionInfo()
```

PriA en cinéticas enzimáticas no tradicionales.

Además de la exploración genómica de PriA se realizaron caracterizaciones experimentales. *i)* Se realizaron cinéticas de PriA en GTP . *ii)* Se avanzó en medir simultáneamente la actividad de PriA sobre ProFAR y PRA.

Es posible que PriA tenga actividad en GTP

Activity was measured fluorometrically in 96-well plates (Nuc 96-Well Optical Botto Plates) in a TECAN infinite M1000 plate reader (excitation at 286 nm and emission at 386 nm)

Preliminary activity assays were performed on an active PriA from *Streptomyces coelicolor* and an inactive mutant D11A.

Enzymes were cloned on coli V68 strains, overexpression were induced and protein were purified.

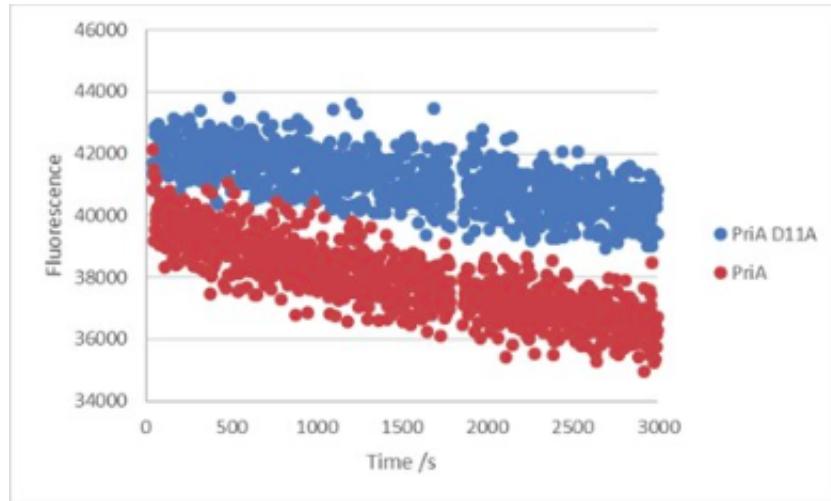


Figure 18:

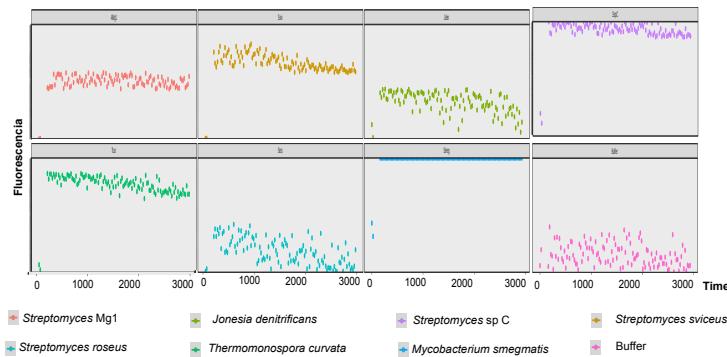


Figure 19:

Enzima	S13	S15	S14	S16	S10	S12	S9	S18	S5	S4	S8	S17	S7	S6	S11	S1
--------	-----	-----	-----	-----	-----	-----	----	-----	----	----	----	-----	----	----	-----	----



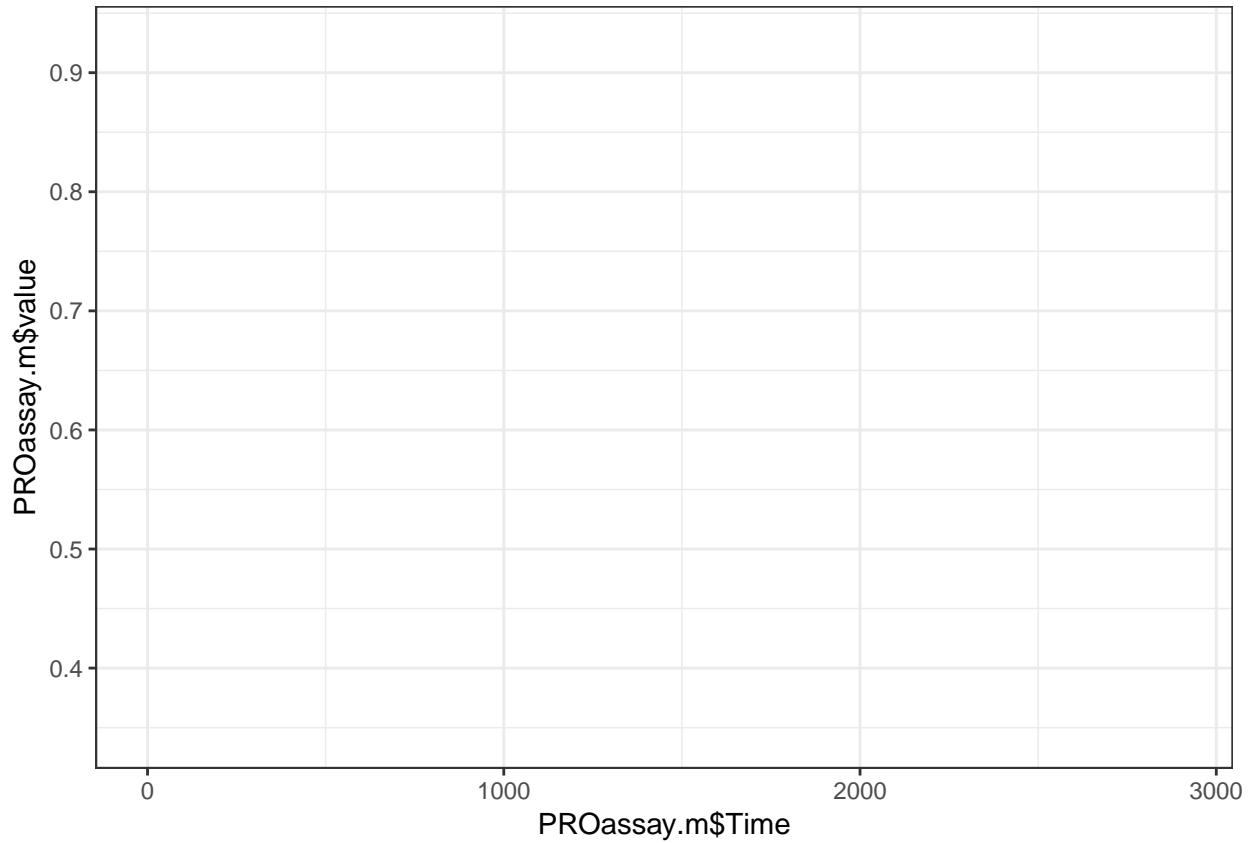
Figure 20:

Cinéticas simultáneas

tle: "TrpF_kinetics"
 thor: "NellySelem"
 te: "May 18, 2017"
 tput: pdf_document

```
#First I saved file as scv tablar separated
#perl -p -i -e 's/,/\t/g' Pra_scoe1.csv
#An cut it to obtain just data
#tail -n +39 Pra_scoe1.csv | head -n-3 > Pra_scoe1.data
#perl -p -i -e 's/^\sNr\|.\|\s\[\w*\]\|/\.\s\[\.*\]\//g' Pra_scoe1.data

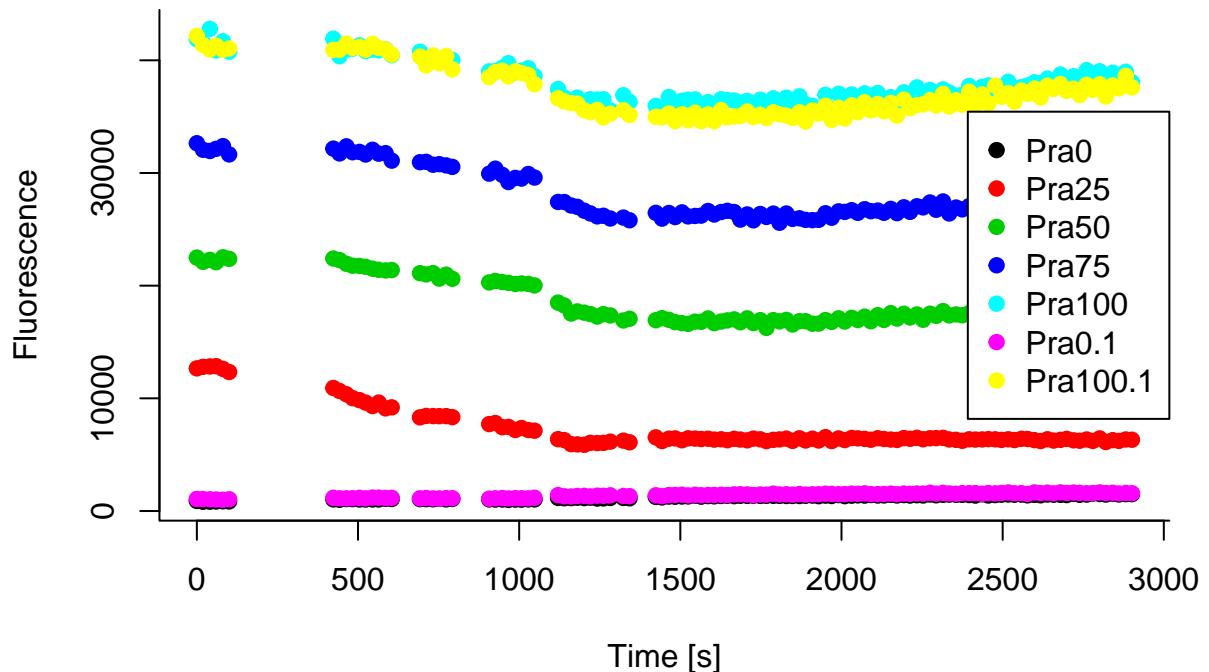

```

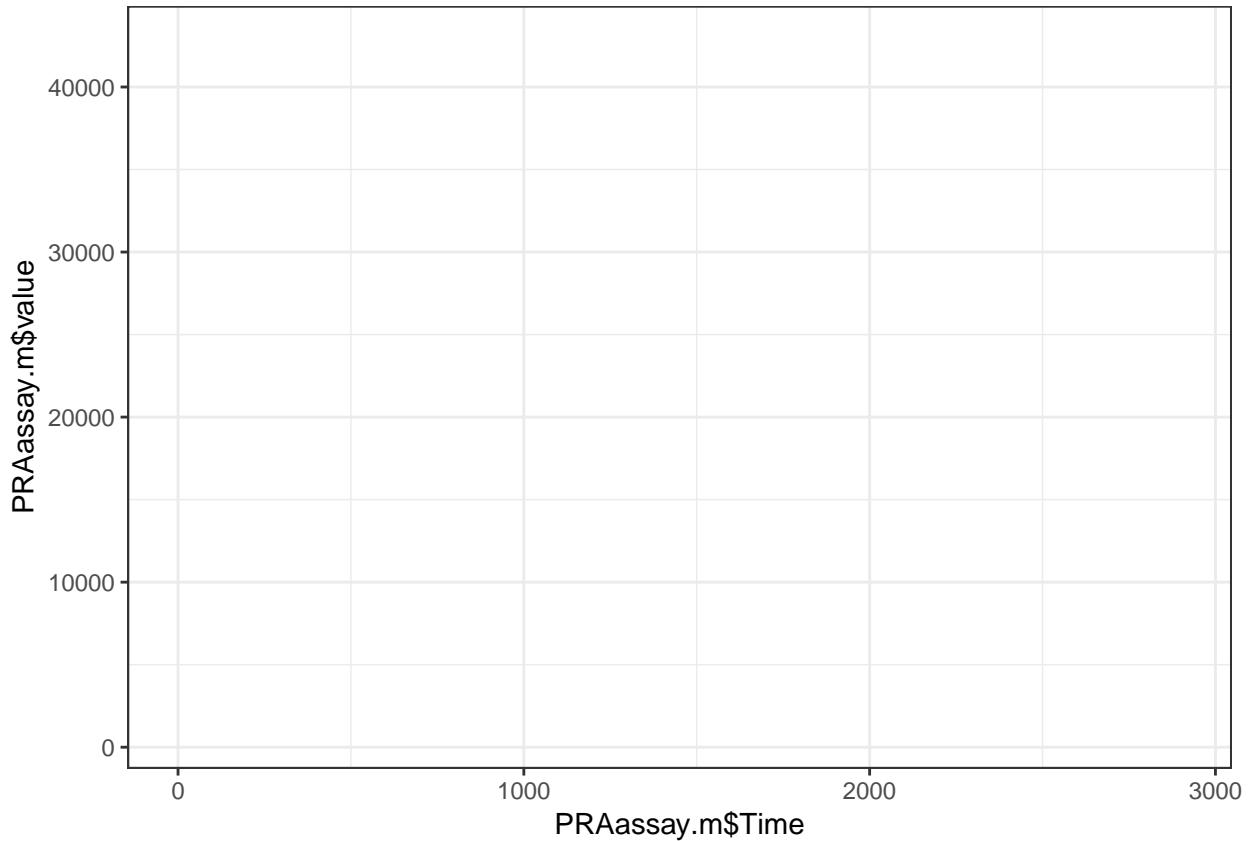
```
qplot(Time,value, data = PROassay.m, colour=variable)+theme_bw() + theme(legend.position = "bottom", legend
```

Warning: Removed 16 rows containing missing values (geom_point).


```
PRAassay.m <- melt(tablePRA,id="Time")
plot(PRAassay.m$Time, PRAassay.m$value, col=PRAassay.m$variable, xlab="Time [s]",ylab="Fluorescence",
par(xpd = TRUE)
legend("right", legend = (unique(PRAassay.m$variable)), col = (unique(PRAassay.m$variable)),pch=19,bg="white")
```

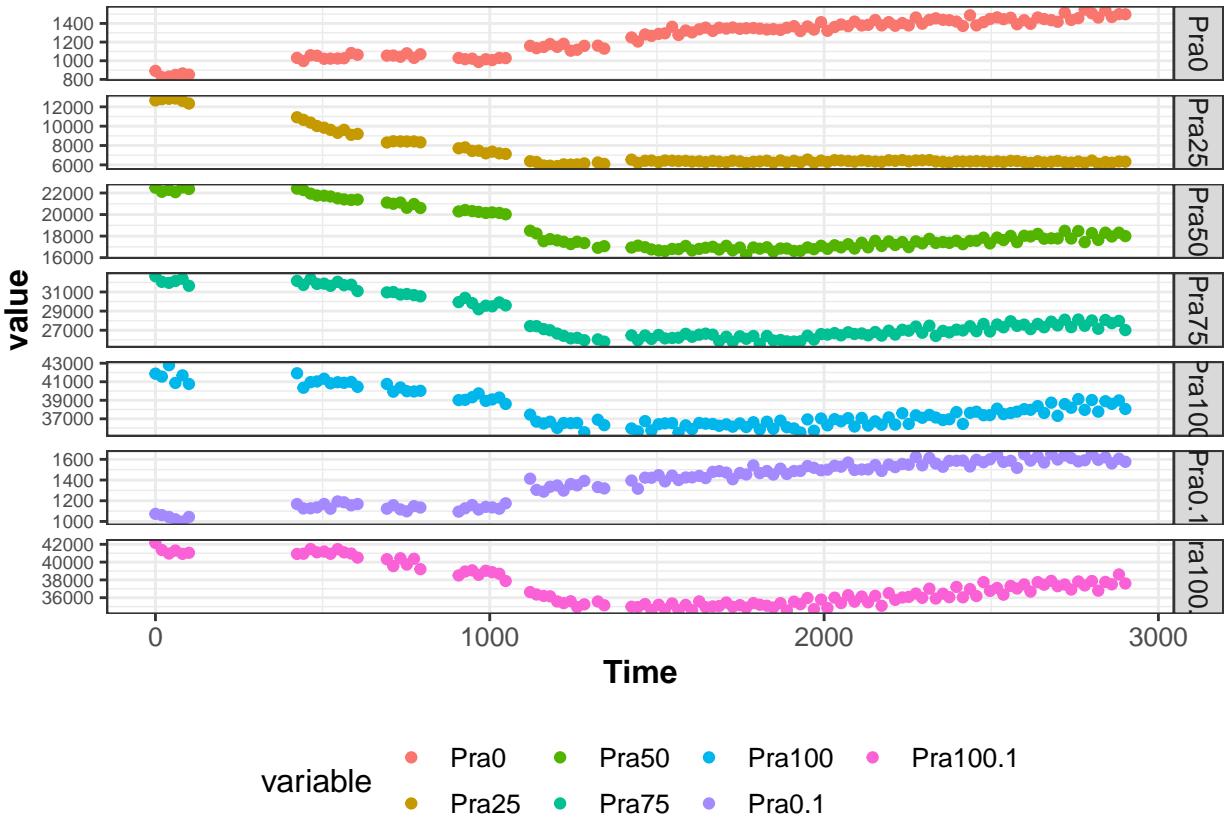


```
ggplot(PRAassay.m, aes(x = PRAassay.m$Time, y = PRAassay.m$value),color="variable") + theme_bw()
```



```
qplot(Time,value, data = PRAassay.m, colour=variable)+theme_bw() + theme(legend.position = "bottom", legend
```

Warning: Removed 7 rows containing missing values (geom_point).



```

##pendientes (slopes) of linear part of the curve
# time when TrpF is added
cuttime<-344.2
maxtime=620

PRAassay2Time<-tablePRA[which(tablePRA$Time >= cuttime & tablePRA$Time <= maxtime),]

## removi columnas que se salian del margen de medicion
#PRAassay2Time<-PRAassay2Time[, !(colnames(PRAassay2Time) %in% c("C62uM"))]

# vector with slopes for each dataset
V0 <- apply(PRAassay2Time, 2, function(x) coefficients(lm(x ~ PRAassay2Time$Time,na.action=na.omit))[2])

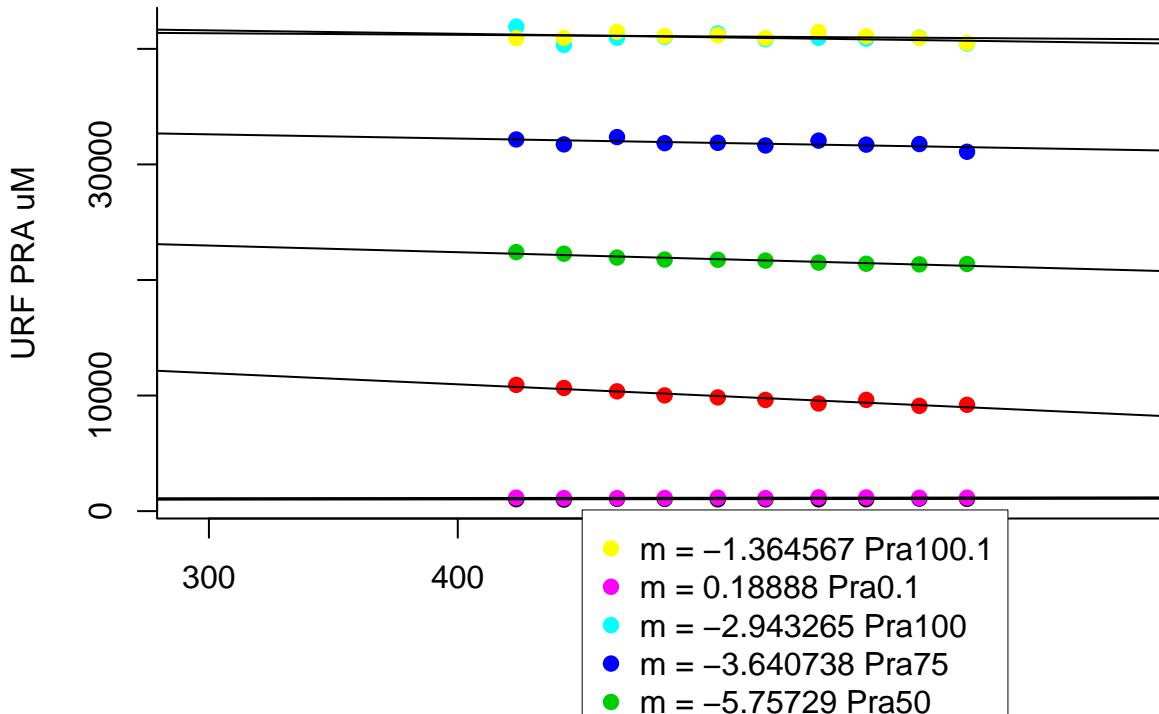
PRAassay2Time.m <- melt(PRAassay2Time,id="Time")
plot(PRAassay2Time.m$Time, PRAassay2Time.m$value, col=PRAassay2Time.m$variable,xlab="Time", ylab="URF P")

V1 <- apply(PRAassay2Time[,2:ncol(PRAassay2Time)], 2, function(x) coefficients(lm(x ~ PRAassay2Time$Time,na.action=na.omit))[2])
apply(V1, 2, function(x) {abline(x, col = PRAassay2Time.m$variable)})

## NULL

par(xpd = TRUE)
slopeText2="m ="
slopeText<-paste(slopeText2,rev(round(V0[-1],digits=6)),rev(names(V0[-1])))
legend(450,100, legend = slopeText, col = rev(unique(PRAassay2Time.m$variable)), pch=19,box.lwd=0, bg="white")

```



```

## get relevant velocities and get correspondent concentrations
## V0 or slopes are on [PRA]uM/s
# slopes stores the slope, with TRUE , FALSE the desired enzyme can be selected, One more than one is t
slopes=(round(V0[-1],digits=6))[c(TRUE)]
concentration<-as.matrix(PRAassay2Time[1,names(slopes)])
#slopes
#concentration
slopes=as.matrix((round(V0[-1],digits=6))[c(TRUE)])
row.names(slopes) <- NULL
slopes<-c(slopes)# plot on r base

slopes

## [1] 0.196328 -9.695510 -5.757290 -3.640738 -2.943265 0.188880 -1.364567
# in concentration I stored the initial substrate concentration
row.names(concentration) <- NULL
concentration<-c(concentration)
slopes

## [1] 0.196328 -9.695510 -5.757290 -3.640738 -2.943265 0.188880 -1.364567
concentration

## [1] 1031 10920 22410 32161 41918 1168 40922
## interpolar michaelis-menden

https://rpubs.com/RomanL/6752
https://davetang.org/muse/2013/05/17/fitting-a-michaelis-mentens-curve-using/
#S <-concentration
## from highest to lower concentration
S_ProFAR <-c(0,25,50,75,100,0,50,100)
v<-slopes

```

```

#v<-slopes[-length(slopes)]
#v <-slopes[c(TRUE, FALSE)]
v<--1*v
S_ProFAR
v
mm <- data.frame(S_ProFAR,v)
model.drm <- drm(v ~ S, data = mm, fct = MM.2())
summary(model.drm)
## first value equal km
## second value = vm
Km=coefficients(model.drm) [1]
Vmax=2*coefficients(model.drm) [2]
Enzyme=2.5 #2.5uM
Kcat=Vmax*Enzyme
Km
Vmax
Kcat
mml <- data.frame(S = seq(0, max(mm$S), length.out = 100))
mml$v <- predict(model.drm, newdata = mml)

## plot on r base
plot(mm,log=' ',xlim=c(0,max(mm$S)), ylim=c(0,max(mm$v)), xlab="Reads", ylab="Transcripts")

##plot on ggplot
ggplot(mm, aes(x = S, y = v)) + theme_bw() + xlab("Concentration [uM]") + ylab("Speed [d[PRA]uM/s]")
ggsave("mm.pdf", width = 6, height = 4)

```