

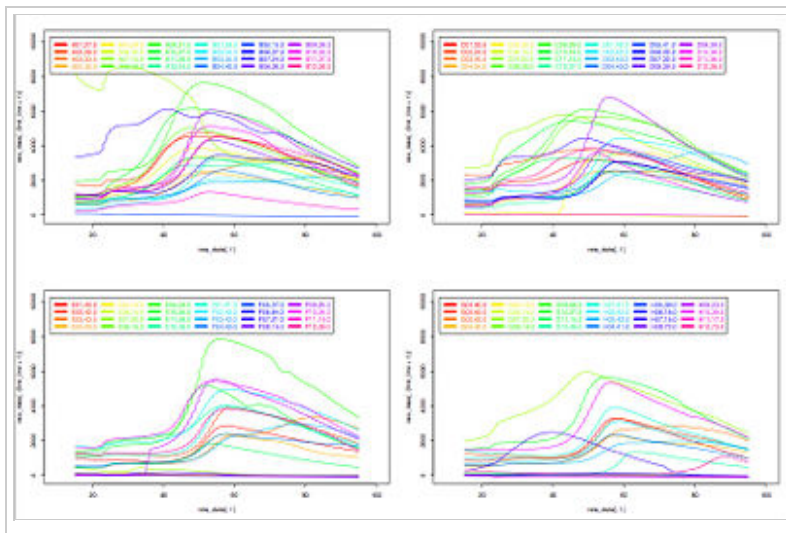
Thermal Shift Assays

From Xtalwiki

Also known as 'Thermofluor' is a way to check the relative stability of proteins and complexes in various buffers to improve stability, by modifying the purification and storage buffers accordingly.

Contents

- 1 Preparation of experiment
 - 1.1 REQUIREMENTS
 - 1.2 PROTOCOL
- 2 Operation of BioRad thermal cycler
- 3 Storage of reagents and controls
- 4 Processing of data
 - 4.1 BioRad software
 - 4.2 Excel
 - 4.3 R



Preparation of experiment

REQUIREMENTS

3-5 μ g of protein per well

0.03 μ l Sypro Orange per well

12.5 μ l solution of 2x buffer screen per well

PROTOCOL

1. Use Biorad PCR 96plate
2. Add 12.5 μ l buffer solution of screen into each well. (work on ice)
3. Make a mastermix of the Sypro orange: Prepare a 100 times dilution of the stock in MilliQ.

Check that the dye is well resuspended in DMSO, make sure you don't have precipitated dye clumps. Otherwise just add few microliters of fresh DMSO and resuspend extensively.

4. Make a mastermix of the target protein in MilliQ or in a VERY poor buffer.
5. Add 5ul of the Sypro Orange mastermix to each well
6. Add 7ul of the protein target mastermix
7. Seal the plate with Biorad's optical seal.
8. Mix the solutions by tapping gently.
9. The samples are now ready for the ThermoFluor experiment.

Operation of BioRad thermal cycler

Make sure the top and bottom part of the BioRad thermal cycler are turned on. Make sure the computer, which is connected to the BioRad Thermal Cycler is also turned on. Open the MyIQ5 biorad program which is located on the desktop of the computer.

Click on the pictogram for Workshop on the left of the program window. Make sure the "melting curve.tmo" is selected. This is a pre-made protocol for measuring the fluorescence of Sypro Orange in a melting curve, from 15 to 95 Celsius.

Open the top part of the Biorad Thermal Cycler by lifting the handle on the lid and gently shifting the lid to the back. You will now see the 96 well heatblock. Place your samples in the block and close the lid by pulling the lid towards you and lowering the handle.

Press Run on the right side of the program window Press Begin Run on the right side of the program window and select the folder in which you want to save the run file. Your run will take approximately 1 hour and 30 minutes.

Storage of reagents and controls

Processing of data

BioRad software

Excel

R

R (<http://www.r-project.org/>) is a software suite for statistical computing and more. Its very handy for analyzing large quantities of data, and more efficient than Excel in the sense that the table of data are not visible all of the time. That lets you focus on the analysis and not on clicking once you grasp the concept. You can download R for free from internet. To process data from the NKi instrument you need to follow these steps:

- In the BioRad MyIQ5 program export or save your data to Excel format in your folder of interest.
- Transport your Excel file to your computer for analysis.

- Open your Excel file and click on the tab RFU.
 - Save the RFU sheet as a comma delimited CSV file on your desktop.
 - Do the same for the -d(RFU)dt sheet.
 - Open the R program.
 - In the R console copy paste the commands (indicated in the boxed script below) in one go or one line at a time to allow you to change the user- and file names to your own when necessary (indicated in red).
- When you want to edit and/or run a command you have already used before, you can simply re-enter the command in the same way as in a Linux terminal, by pressing the arrows up or down
- Run the commands by pressing enter.

The following code tells R to read the data from the CSV files

```
raw_data <- read.csv("C:/Documents and settings/username/Desktop/yourRFUfile.csv")
grad_data <- read.csv("C:/Documents and settings/username/Desktop/your-dRFUdtfile.csv")
```

In the box below there is a very simple function to make the scaled plots for the RFUs and the derivative and choose as Tm the maximum of the derivative function, after 25 degrees to avoid glitches. To run this simply copy paste it into your R console and press enter.

```
find.tm <- function(temp=temp, I=I, grad=grad, well=well) {
Igrad <- matrix(1:322,nc=2)
Igrad[,1]=I
Igrad[,2]=-grad
scaled_data<-scale(Igrad)
plot(x=temp, y=scaled_data[,1], type='p', col='red', xlab="", ylab="",ylim=c(-5,5))
lines(x=temp, y=scaled_data[,2], type='l', col='green', lwd=2,xlab="", ylab="")
title(main=well)
tm.s=temp[which.max(Igrad[24:150,2])+23]
title(sub=sprintf("Tm = %4.1f", tm.s, cex.sub=1.2))
return(tm.s)
}
```

This here will call the function above to make individual plots in A3 format and put them in a PDF file

```
pdf (file="C:/Documents and settings/username/Desktop/YourThermoGraphs1.pdf", width=30, height=21,point
layout(matrix(data=1:96, nrow=8, ncol=12, byrow=TRUE))
tma<-matrix(nrow=12,ncol=8)
for (i in 2:97) {
try(expr=tma[i-1]<-find.tm(temp=raw_data[,1], I=raw_data[,i], grad=grad_data[,i], well=names(raw_data)
}
dev.off()
```

This will make a PDF file with a Tm coloring of the 96-plate "image' Yellow is Stable, Red is not so stable

```
pdf (file="C:/Documents and settings/username/Desktop/YourThermoGraphs2.pdf", width=6, height=5,paper=
tmaplot<-matrix(nrow=12,ncol=8,data=0)
for (i in 1:8) {
tmaplot[,9-i]=tma[,i]
}
image(tmaplot)
dev.off()
```

And this will make an A4 with two classic plots (2x48) of the primary data its fairly easy to modify so as to get any data you want without compulsive clicking.

```
pdf (file="C:/Documents and settings/username/Desktop/YourThermoGraphs3.pdf", width=8, height=12,points
layout(matrix(data=1:2, nrow=2, ncol=1, byrow=TRUE))
xlim = c(min(raw_data[,1]),max(raw_data[,1]))
ylim = c(min(raw_data[,2:97]),max(raw_data[,2:97])*1.2)
y.leg=max(raw_data[,2:97])*1.2
x.leg=8
ccol=c(rainbow(48, start=0, end=0.99))
# Make the first line plot
plot(x=raw_data[,1],y=raw_data[,2],type='l', col=(ccol[1]), ylim=ylim, xlim=c(10,100) )
#and now add 47 lines
for (i in 3:49) {
lines(x=raw_data[,1],y=raw_data[,i],type='l',col=(ccol[i-1]))
}
#Set the legends
leg.txt=sprintf("%s,%3.1f",names(raw_data[2:49]),tma[1:48])
#and print them in 8 columns
legend(x.leg,y.leg,leg.txt,text.col=ccol,lty=1,lwd=4,col=ccol,ncol=8)
#Repeat for the second bunch of 48!
plot(x=raw_data[,1],y=raw_data[,49],type='l', col=(ccol[49]), ylim=ylim, xlim=c(10,100) )
for (i in 50:97) {
lines(x=raw_data[,1],y=raw_data[,i],type='l',col=(ccol[i-48]))
}
leg.txt=sprintf("%s,%3.1f",names(raw_data[50:97]),tma[49:96])
legend(x.leg,y.leg,leg.txt,text.col=ccol,lty=1,lwd=4,col=ccol,ncol=8)
dev.off()
```

This one will produce 4 panels with line graphs in landscape mode, its a very small modification of the script above.

```
pdf (file="C:/Documents and settings/username/Desktop/YourThermoGraphs4.pdf", width=12, height=8,points
layout(matrix(data=1:4, nrow=2, ncol=2, byrow=TRUE))
xlim = c(min(raw_data[,1]),max(raw_data[,1]))
ylim = c(min(raw_data[,2:97]),max(raw_data[,2:97])*1.2)
y.leg=max(raw_data[,2:97])*1.2
x.leg=8
ccol=c(rainbow(24, start=0, end=0.9))
#
first_line=1
#
# Make the first line plot
plot(x=raw_data[,1],y=raw_data[(first_line+1)],type='l', col=(ccol[1]), ylim=ylim, xlim=c(10,100) )
#and now add 23 lines
l1=first_line+2
l2=first_line+24
for (i in l1:l2) {
lines(x=raw_data[,1],y=raw_data[,i],type='l',col=(ccol[i-first_line]))
}
#Set the legends
leg.txt=sprintf("%s,%3.1f",names(raw_data[(l1-1):l2]),tma[(first_line):(first_line+23)])
#and print them in 8 columns
legend(x.leg,y.leg,leg.txt,text.col=ccol,lty=1,lwd=4,col=ccol,ncol=6)
#
first_line=25
plot(x=raw_data[,1],y=raw_data[(first_line+1)],type='l', col=(ccol[1]), ylim=ylim, xlim=c(10,100) )
l1=first_line+2
l2=first_line+24
for (i in l1:l2) {
lines(x=raw_data[,1],y=raw_data[,i],type='l',col=(ccol[i-first_line]))
}
leg.txt=sprintf("%s,%3.1f",names(raw_data[(l1-1):l2]),tma[(first_line):(first_line+23)])
legend(x.leg,y.leg,leg.txt,text.col=ccol,lty=1,lwd=4,col=ccol,ncol=6)
#
first_line=49
```

```

plot(x=raw_data[,1],y=raw_data[(first_line+1)],type='l', col=(ccol[1]), ylim=yylim, xlim=c(10,100) )
l1=first_line+2
l2=first_line+24
for (i in l1:l2) {
lines(x=raw_data[,1],y=raw_data[,i],type='l',col=(ccol[i-first_line]))
}
leg.txt=sprintf("%s,%3.1f",names(raw_data[(l1-1):l2]),tma[(first_line):(first_line+23)])
legend(x.leg,y.leg,leg.txt,text.col=ccol,lty=1,lwd=4,col=ccol,ncol=6)
#
first_line=73
plot(x=raw_data[,1],y=raw_data[(first_line+1)],type='l', col=(ccol[1]), ylim=yylim, xlim=c(10,100) )
l1=first_line+2
l2=first_line+24
for (i in l1:l2) {
lines(x=raw_data[,1],y=raw_data[,i],type='l',col=(ccol[i-first_line]))
}
leg.txt=sprintf("%s,%3.1f",names(raw_data[(l1-1):l2]),tma[(first_line):(first_line+23)])
legend(x.leg,y.leg,leg.txt,text.col=ccol,lty=1,lwd=4,col=ccol,ncol=6)
dev.off()

```

Note that the input files should look like that:

```

, A01, A02, ....
15, 1000, 1000, ...
15.5, 1001, 990, ...

```

with a top line with headers, and then the values, where the first column is the temperature.

Also note that you need to have exactly 1+96 columns (temperature and 96 measurements) and 161 rows (161 step temperatures).

Retrieved from "http://xtal.nki.nl:8008/wiki/index.php/Thermal_Shift_Assays"

- This page was last modified 10:49, 1 October 2008.