Arpan's Hypothesis (19054)

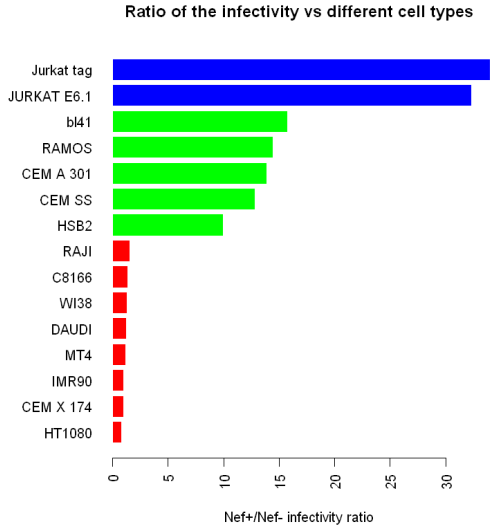
HIV-1 Nef factor induces high expression of HIRA (Histone cell cycle regulator) gene to increase the viral genome integration efficiency inside the infected cell.

Installing packages and libraries

In [13]:
```
# install.packages("reshape")
# install.packages("BiocManager")
# install.packages("ggrepel")
# install.packages("corrplot")
# install.packages("pheatmap")
# BiocManager::install("ComplexHeatmap")
# BiocManager::install("AnnotationDbi")
# BiocManager::install("org.Hs.eg.db")
# BiocManager::install('PCAtools')
# install.packages("rlang")
# install.packages("gplots")
# install.packages("factoextra")
# install.packages("corrr")
# install.packages("ggcorrplot")
# install.packages("FactoMineR")
library(corrr)
library(ggcorrplot)
library(FactoMineR)
library(PCAtools)
library(reshape)
library(org.Hs.eg.db)
library(tibble)
library(dplyr)
library(ggplot2)
library(gplots)
library(ggrepel)
library(pheatmap)
library(rlang)
library(factoextra)
library(vctrs)
```

reading infect.txt and plotting

In [14]:
```
# ------------Code for Ratio of the infectivity vs different cell types plot--------
setwd("D:/Study/SEMESTER 8/Biostats/datasets")
read.delim("infect.txt", fill=TRUE, header=FALSE ) -> infect_data
infect_data <- infect_data[order(infect_data$V2),]
infect_ratio <- infect_data$V2
names(infect_ratio) <- infect_data$V3

give_color <- function(value){
  if (value < 5) return("red")
  else if ( value < 25) return("green")
  else return("blue")
}
par(mar = c(6, 8, 4, 4)) # for margin around plot

color_vector <- unlist(lapply(infect_data$V2, give_color))
infect_data$V4 <- color_vector

barplot(infect_ratio, horiz = TRUE,
        las=2, xlab = "Nef+/Nef- infectivity ratio", col=color_vector,
        main = "Ratio of the infectivity vs different cell types",
        border = 0)
```



Reading htseq files and merging to dataframe

In [15]:
```
expression_df <- read.delim("SRR2166624.htseq",header = FALSE,col.names = c("ID","SRR2166624"))

# appending columns to expression_df from other htseq files using a for loop
for (i in 25:38){
  column_name <- paste("SRR21666",as.character(i), sep="")
  filename <- paste("SRR21666",as.character(i),".htseq", sep="") # generating the given filenames using a loop
  temp_df <- read.delim(filename,header = FALSE,col.names = c("ID",column_name))
  expression_df[,column_name] <- temp_df[,column_name] # appending new columns to expression_df
}
# Changing column names to Cell line names from infect.txt file
colnames(expression_df) <- c( "ID",infect_data[order(infect_data$V1,decreasing=FALSE),]$V3)
# getting rid of summary values in the end of the dataframe
clean_df <- expression_df[1:58302,]
```

In [16]:
```
head(clean_df,10)
```

A data.frame: 10 × 16

	ID	MT4	HSB2	HT1080	RAJI	CEM SS	DAUDI	C8166	RAMOS	IMR90	CEM A 301	CEM X 174	WI38	JURKAT E6.1	bl41	Jurkat tag
	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	ENSG00000000003	0	64	71	0	0	0	2	0	963	0	0	572	0	0	4
2	ENSG00000000005	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	ENSG00000000419	535	707	535	888	706	424	655	542	1021	29	415	645	771	410	1128
4	ENSG00000000457	643	643	165	604	490	475	292	362	470	13	109	496	1630	477	887
5	ENSG00000000460	885	826	143	648	808	654	354	555	441	22	140	317	2339	593	1428
6	ENSG00000000938	9	8	1	2977	3	1188	14	134	0	0	24	0	74	10	16
7	ENSG00000000971	1	228	5	0	16	0	0	0	971	2	0	2560	4	0	4
8	ENSG00000001036	174	12	334	0	4	4	2	2	2766	0	8	2069	55	12	15
9	ENSG00000001084	549	635	79	491	987	769	298	388	1231	26	105	1855	1319	769	1282
10	ENSG00000001167	586	1015	232	1238	1116	616	994	535	1321	36	186	1063	3303	454	1657

Cleaning up data and adding gene names column

In [17]:
```
clean_df$gene_names <- mapIds(org.Hs.eg.db, keys=clean_df$ID, keytype="ENSEMBL", column="SYMBOL") # adding new column with gene names
clean_df[is.na(clean_df$gene_names),17] <- "Unknown" # unknown gene names are converted to NA # adding new column gene names from accession num

cdf <- as.data.frame(sapply(clean_df[,c(-1,-17)], as.numeric))
cdf %>% mutate(zc = rowSums(. == 0)) -> cdf # counting zeros row wise
nonzero_cdf <- cdf[cdf$zc < 7,][-16] # filtering genes with non-zero expression in at least 50% cell lines

head(nonzero_cdf,10)
```

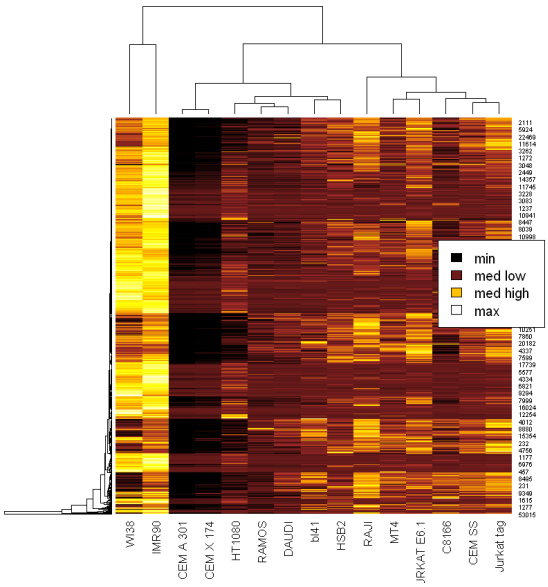
```
'select()' returned 1:many mapping between keys and columns
```

A data.frame: 10 × 15

	MT4	HSB2	HT1080	RAJI	CEM SS	DAUDI	C8166	RAMOS	IMR90	CEM A 301	CEM X 174	WI38	JURKAT E6.1	bl41	Jurkat tag
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
3	535	707	535	888	706	424	655	542	1021	29	415	645	771	410	1128
4	643	643	165	604	490	475	292	362	470	13	109	496	1630	477	887
5	885	826	143	648	808	654	354	555	441	22	140	317	2339	593	1428
6	9	8	1	2977	3	1188	14	134	0	0	24	0	74	10	16
7	1	228	5	0	16	0	0	0	971	2	0	2560	4	0	4
8	174	12	334	0	4	4	2	2	2766	0	8	2069	55	12	15
9	549	635	79	491	987	769	298	388	1231	26	105	1855	1319	769	1282
10	586	1015	232	1238	1116	616	994	535	1321	36	186	1063	3303	454	1657
11	44	144	58	266	19	67	79	63	489	0	2	318	50	131	51
12	359	638	346	708	234	241	438	271	5274	3	16	3876	1072	411	877

Plotting overall expression of significant genes profile as heatmap

In [18]:
```
my_colors <- colorRampPalette(c("black", "black","brown","orange","yellow","white"))
heatmap(as.matrix(nonzero_cdf[order(nonzero_cdf$IMR90, decreasing = TRUE), ][0:1000,0:15]), col=my_colors(1000))
legend(x="right", legend=c("min", "med low", "med high", "max"),fill=my_colors(4))
```



Normalizing data -> getting RPM values

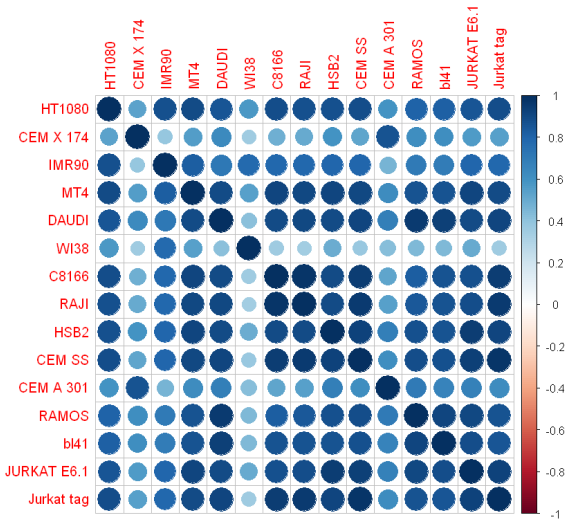
In [19]:
```
rpm_df <- clean_df

for (name in (colnames(rpm_df)[c(-17,-1)])){
col_total = sum(as.numeric(rpm_df[,name]))
current_col = rpm_df[,name]
rpm_df[,name] <- unlist(lapply(current_col, FUN = function(x) x*1000000/col_total))
}
rpm_df$name <- NULL
rpm_df <- rpm_df[,c('ID',arrange(infect_data,V2)$V3,'gene_names')]
head(rpm_df,10)
```

A data.frame: 10 × 17

	ID	HT1080	CEM X 174	IMR90	MT4	DAUDI	WI38	C8166	RAJI	HSB2	CEM SS	CEM A 301	RAMOS
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	ENSG00000000003	9.9611068	0.000000	35.44316	0.00000000	0.000000	24.70044	0.2057234	0.00000	4.81296698	0.0000000	0.000000	0.0000000
2	ENSG00000000005	0.0000000	0.000000	0.00000	0.00000000	0.000000	0.00000	0.0000000	0.00000	0.07520261	0.0000000	0.000000	0.0000000
3	ENSG00000000419	75.0590441	162.591306	37.57784	40.02970429	44.760829	27.85277	67.3744100	49.63045	53.16824456	57.0600174	94.991336	65.0361629
4	ENSG00000000457	23.1490510	42.704704	17.29832	48.11046702	50.144797	21.41857	30.0356148	33.75765	48.35527758	39.6025616	42.582323	43.4374372
5	ENSG00000000460	20.0625109	54.850079	16.23098	66.21736129	69.041468	13.68888	36.4130399	36.21681	62.11735502	65.3038159	72.062393	66.5960708
6	ENSG00000000938	0.1402973	9.402871	0.00000	0.67339689	125.414777	0.00000	1.4400637	166.38496	0.60162087	0.2424647	0.000000	16.0790513
7	ENSG00000000971	0.7014864	0.000000	35.73760	0.07482188	0.000000	110.54743	0.0000000	0.00000	17.14619485	1.2931449	6.551127	0.0000000
8	ENSG00000001036	46.8592911	3.134290	101.80246	13.01900663	0.422272	89.34478	0.2057234	0.00000	0.90243131	0.3232862	0.000000	0.2399858
9	ENSG00000001084	11.0834850	41.137559	45.30688	41.07721057	81.181787	80.10371	30.6527850	27.44206	47.75365671	79.7708741	85.164646	46.5572531
10	ENSG00000001167	32.5489687	72.872248	48.61932	43.84562002	65.029884	45.90309	102.2445245	69.19200	76.33064812	90.1968546	117.920279	64.1962124

Correlation Plot

In [21]:
```
d <- cor(rpm_df[,c(-1,-17)])
library(corrplot)
suppressWarnings(corrplot(d))
```



Establishing correlation of infect ratio with expression data

In [24]:
```
find_cor <- function(onerow, methodname){
sorted_infect_ratio <- arrange(infect_data,V2)$V2
cor_value <- suppressWarnings(cor.test(sorted_infect_ratio,as.numeric(as.vector(unlist(onerow))[c(-1,-17)]), method=methodname))
return(as.numeric(cor_value$estimate))
}

find_cor_p_val <- function(onerow, methodname){
sorted_infect_ratio <- arrange(infect_data,V2)$V2
cor_value <- suppressWarnings(cor.test(sorted_infect_ratio,as.numeric(as.vector(unlist(onerow))[c(-1,-17)]), method=methodname))
return(as.numeric(cor_value$p.value))
}
```

HIRA is not normally distributed

In [25]:
```
shapiro.test(as.vector(unlist(rpm_df[rpm_df$gene_names=="HIRA",][2:16])))
```

```
        Shapiro-Wilk normality test

data:  as.vector(unlist(rpm_df[rpm_df$gene_names == "HIRA", ][2:16]))
W = 0.84814, p-value = 0.01635
```

Spearman correlation test [non-parametric] & Multiple testing correction

In [26]:
```
result_df <- rpm_df
result_df$cor_with_infect_ratio <- as.vector(apply(rpm_df,1,find_cor, methodname="spearman"))
result_df$cor_with_p_val <- as.vector(apply(rpm_df,1,find_cor_p_val, methodname="spearman"))
result_df$cor_with_p_val_adjusted <- p.adjust(result_df$cor_with_p_val, method = "bonferroni")
spearman_result_df_sorted <- arrange(result_df, desc(cor_with_infect_ratio))
spearman_result_df_sorted <- subset(spearman_result_df_sorted, !is.na(cor_with_infect_ratio))
spearman_result_df_sorted <- spearman_result_df_sorted[spearman_result_df_sorted$gene_names != "Unknown",]
head(spearman_result_df_sorted,10)
```

A data.frame: 10 × 20

	ID	HT1080	CEM X 174	IMR90	MT4	DAUDI	WI38	C8166	RAJI	HSB2	CEM SS	CEM A 301	RAM
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dl
1	ENSG00000130921	11.7849714	22.331818	19.6906420	21.2494131	26.180862	27.3345804	19.8523071	30.8513600	27.674560	40.814885	39.306760	29.998
2	ENSG00000137713	54.1547496	70.521530	66.6169383	59.4085705	77.909179	60.6283585	78.4834730	77.9108621	79.263550	98.844761	160.502602	175.669
3	ENSG00000174010	20.4834027	31.734689	21.4940840	33.8943104	42.227198	20.9867395	24.8925301	43.2589722	56.702767	47.280609	68.786830	75.475
4	ENSG00000100084	7.2954585	7.835726	7.3241827	16.5356349	19.530079	8.1183272	7.9203505	13.0224038	15.642143	19.397173	19.653380	20.398
5	ENSG00000116906	56.1189115	52.891148	50.2387408	71.0807833	75.797820	57.1737512	88.9753659	76.9607297	75.879432	83.650309	94.991336	101.154
6	ENSG00000162521	134.2644957	358.876232	159.5862124	318.4419092	450.986470	166.2961600	226.2957282	377.7614901	362.476575	385.114706	455.303301	614.963
9	ENSG00000143179	80.8112325	72.480462	84.7249679	100.8598904	38.004478	81.0969069	109.8562899	114.1835663	138.824016	134.163780	222.738305	103.553
10	ENSG00000225171	0.0000000	0.000000	0.1840247	0.9726844	1.055680	0.2590955	0.4114468	0.8383522	1.353647	1.454788	3.275563	1.319
11	ENSG00000143363	7.4357558	4.309649	4.8214469	7.7066533	14.251679	10.0183612	6.4802868	12.3517220	12.408430	19.316351	26.204507	16.199
12	ENSG00000152475	0.8417837	0.000000	2.4659309	1.7957251	2.322496	0.7772866	1.9543722	2.4032762	4.136143	5.657509	3.275563	2.999

Kendall correlation test [non-parametric] & Multiple testing correction

In [27]:
```
result_df <- rpm_df
result_df$cor_with_infect_ratio <- as.vector(apply(rpm_df,1,find_cor, methodname="kendall"))
result_df$cor_with_p_val <- as.vector(apply(rpm_df,1,find_cor_p_val, methodname="kendall"))
result_df$cor_with_p_val_adjusted <- p.adjust(result_df$cor_with_p_val, method = "bonferroni")
kendall_result_df_sorted <- arrange(result_df, desc(cor_with_infect_ratio))
```

```
kendall_result_df_sorted <- subset(kendall_result_df_sorted, !is.na(cor_with_infect_ratio))
kendall_result_df_sorted <- kendall_result_df_sorted[kendall_result_df_sorted$gene_names != "Unknown",]
head(kendall_result_df_sorted,10)
```

A data.frame: 10 × 20

	ID	HT1080	CEM X 174	IMR90	MT4	DAUDI	WI38	C8166	RAJI	HSB2	CEM SS	CEM A 301	
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	ENSG00000100084	7.295458	7.835726	7.32418272	16.5356349	19.530079	8.11832721	7.9203505	13.02240379	15.6421427	19.397173	19.653380	20.3
2	ENSG00000137713	54.154750	70.521530	66.61693829	59.4085705	77.909179	60.62835853	78.4834730	77.91086215	79.2635499	98.844761	160.502602	175.6
3	ENSG00000162521	134.264496	358.876232	159.58621238	318.4419092	450.986470	166.29616003	226.2957282	377.76149015	362.4765753	385.114706	455.303301	614.9
4	ENSG00000174010	20.483403	31.734689	21.49408396	33.8943104	42.227198	20.98673949	24.8925301	43.25897224	56.7027672	47.280609	68.786830	75.4
5	ENSG00000130921	11.784971	22.331818	19.69064198	21.2494131	26.180862	27.33458045	19.8523071	30.85136005	27.6745601	40.814885	39.306760	29.9
6	ENSG00000178502	15.152106	17.238596	18.03441976	18.7802912	34.731870	15.71846332	21.4980942	22.07660728	23.0872010	33.379302	36.031196	34.4
8	ENSG00000143179	80.811233	72.480462	84.72496792	100.8598904	38.004478	81.09690692	109.8562899	114.18356626	138.8240162	134.163780	222.738305	103.5
10	ENSG00000225171	0.000000	0.000000	0.18402469	0.9726844	1.055680	0.25909555	0.4114468	0.83835218	1.3536470	1.454788	3.275563	1.3
12	ENSG00000230524	0.000000	0.000000	0.03680494	0.0000000	0.105568	0.04318259	0.0000000	0.05589015	0.1504052	1.050680	3.275563	0.1
13	ENSG00000116906	56.118911	52.891148	50.23874075	71.0807833	75.797820	57.17375120	88.9753659	76.96072968	75.8794325	83.650309	94.991336	101.1

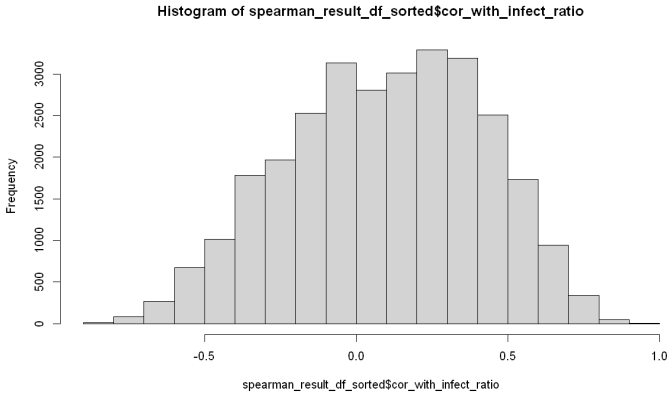
Pearson correlation test [non-parametric] & Multiple testing correction

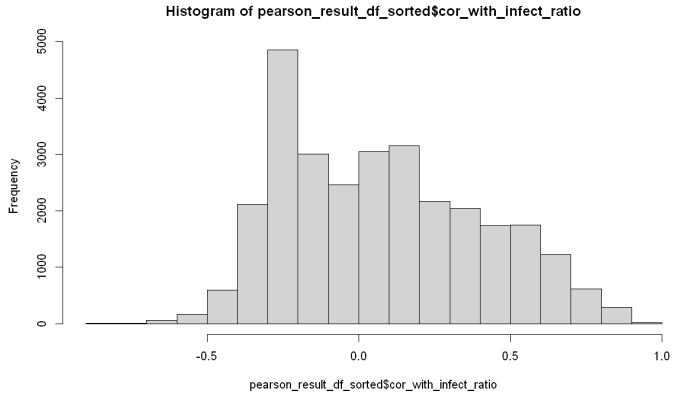
In [28]:
```
result_df <- rpm_df
result_df$cor_with_infect_ratio <- as.vector(apply(rpm_df,1,find_cor, methodname="pearson"))
result_df$cor_with_p_val <- as.vector(apply(rpm_df,1,find_cor_p_val, methodname="pearson"))
result_df$cor_with_p_val_adjusted <- p.adjust(result_df$cor_with_p_val, method = "bonferroni")
pearson_result_df_sorted <- arrange(result_df, desc(cor_with_infect_ratio))
pearson_result_df_sorted <- subset(pearson_result_df_sorted, !is.na(cor_with_infect_ratio))
pearson_result_df_sorted <- pearson_result_df_sorted[pearson_result_df_sorted$gene_names != "Unknown",]
head(pearson_result_df_sorted,10)
```

A data.frame: 10 × 20

	ID	HT1080	CEM X 174	IMR90	MT4	DAUDI	WI38	C8166	RAJI	HSB2	CEM SS	CEM A 301	RAMOS
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	ENSG00000164300	18.6595381	10.9700158	20.79479013	8.0059409	3.800448	31.48010923	3.6001593	6.7068174	201.843803	128.910379	189.982672	134.032092
2	ENSG00000136104	18.6595381	97.9465698	14.02268148	104.4513405	113.802297	12.82522969	20.5723389	57.7904099	165.445740	182.171783	186.707109	162.110435
3	ENSG00000152580	2.8059456	0.7835726	0.07360988	9.6520222	0.105568	0.47500851	0.2057234	4.8624426	20.154299	8.971193	9.826690	12.959235
6	ENSG00000100084	7.2954585	7.8357256	7.32418272	16.5356349	19.530079	8.11832721	7.9203505	13.0224038	15.642143	19.397173	19.653380	20.398796
7	ENSG00000144893	14.0297279	0.7835726	0.66248889	68.3871957	0.211136	1.12274738	1.7486488	36.3285943	157.850276	73.628436	104.818026	85.794938
8	ENSG00000144736	36.3369952	39.9622005	21.56769383	19.3788662	37.159934	23.36178202	28.1841043	48.7920966	74.751393	103.936519	65.511266	58.316559
9	ENSG00000196839	23.7102401	92.8533481	4.41659259	0.1496438	17.524287	6.04556282	0.2057234	59.5788946	114.458371	116.867968	157.227039	70.675830
10	ENSG00000172995	0.1402973	0.3917863	0.07360988	0.0000000	1.583520	0.04318259	0.0000000	0.3353409	0.000000	49.058683	85.164646	2.159873
11	ENSG00000152475	0.8417837	0.0000000	2.46593086	1.7957251	2.322496	0.77728665	1.9543722	2.4032762	4.136143	5.657509	3.275563	2.999823
12	ENSG00000163930	59.0651543	52.1075751	66.80096298	103.4038342	100.923002	62.74430551	66.1400696	69.9744616	119.421743	160.673250	117.920279	107.633650

In [29]:
```
options(repr.plot.width=20, repr.plot.height=6)
par(mfrow=c(1,2))
hist(spearman_result_df_sorted$cor_with_infect_ratio)
hist(pearson_result_df_sorted$cor_with_infect_ratio)
```





In [30]:
```
rpm_df[rpm_df$gene_names=="SERINC5",]
```

A data.frame: 1 × 17

	ID	HT1080	CEM X 174	IMR90	MT4	DAUDI	WI38	C8166	RAJI	HSB2	CEM SS	CEM A 301	RAMOS	bl41	JURKAT E6.1	Ju
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<c
11315	ENSG00000164300	18.65954	10.97002	20.79479	8.005941	3.800448	31.48011	3.600159	6.706817	201.8438	128.9104	189.9827	134.0321	115.0093	397.5008	322.9

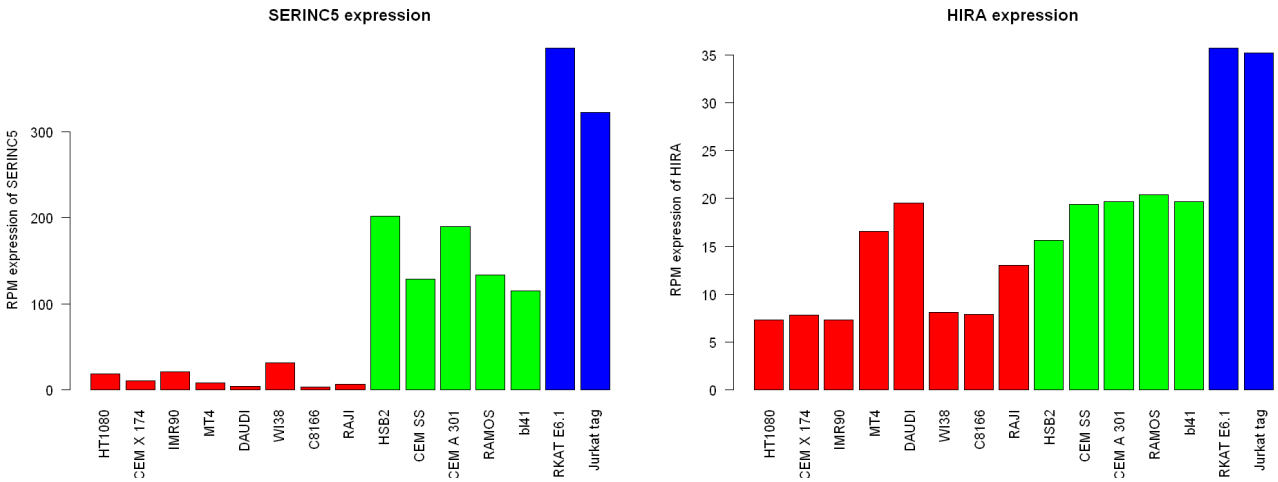
In [31]:
```
rpm_df[rpm_df$gene_names=="HIRA",]
```

A data.frame: 1 × 17

	ID	HT1080	CEM X 174	IMR90	MT4	DAUDI	WI38	C8166	RAJI	HSB2	CEM SS	CEM A 301	RAMOS	bl41	JURKAT E6.1	Jurkat tag
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2184	ENSG00000100084	7.295458	7.835726	7.324183	16.53563	19.53008	8.118327	7.92035	13.0224	15.64214	19.39717	19.65338	20.3988	19.69993	35.71868	35.17124

Plotting SERINC5 expression

In [32]:
```
options(repr.plot.width=16, repr.plot.height=6)
par(mfrow=c(1,2))
SERINC5_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="SERINC5",][,c(-1,-17)]))
names(SERINC5_expression) <- colnames(rpm_df)[c(-1,-17)]
barplot(SERINC5_expression,las=2, col=arrange(infect_data,V2)$V4, main="SERINC5 expression", ylab="RPM expression of SERINC5")
HIRA_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="HIRA",][,c(-1,-17)]))
names(HIRA_expression) <- colnames(rpm_df)[c(-1,-17)]
barplot(HIRA_expression,las=2, col=arrange(infect_data,V2)$V4, main="HIRA expression", ylab="RPM expression of HIRA")
```



In [33]:
```
high_Nef_dependent_cell_lines_HIRA_expression = as.vector(unlist(
    rpm_df[rpm_df$gene_names=="HIRA",][,c('HSB2','CEM SS','CEM A 301','RAMOS','bl41','JURKAT E6.1')]))

low_Nef_dependent_cell_lines_HIRA_expression = as.vector(
    unlist(rpm_df[rpm_df$gene_names=="HIRA",][,c('HT1080','CEM X 174','IMR90','MT4','DAUDI','WI38','C8166','RAJI')]))

wilcox.test(low_Nef_dependent_cell_lines_HIRA_expression,high_Nef_dependent_cell_lines_HIRA_expression, alternative="less")
```

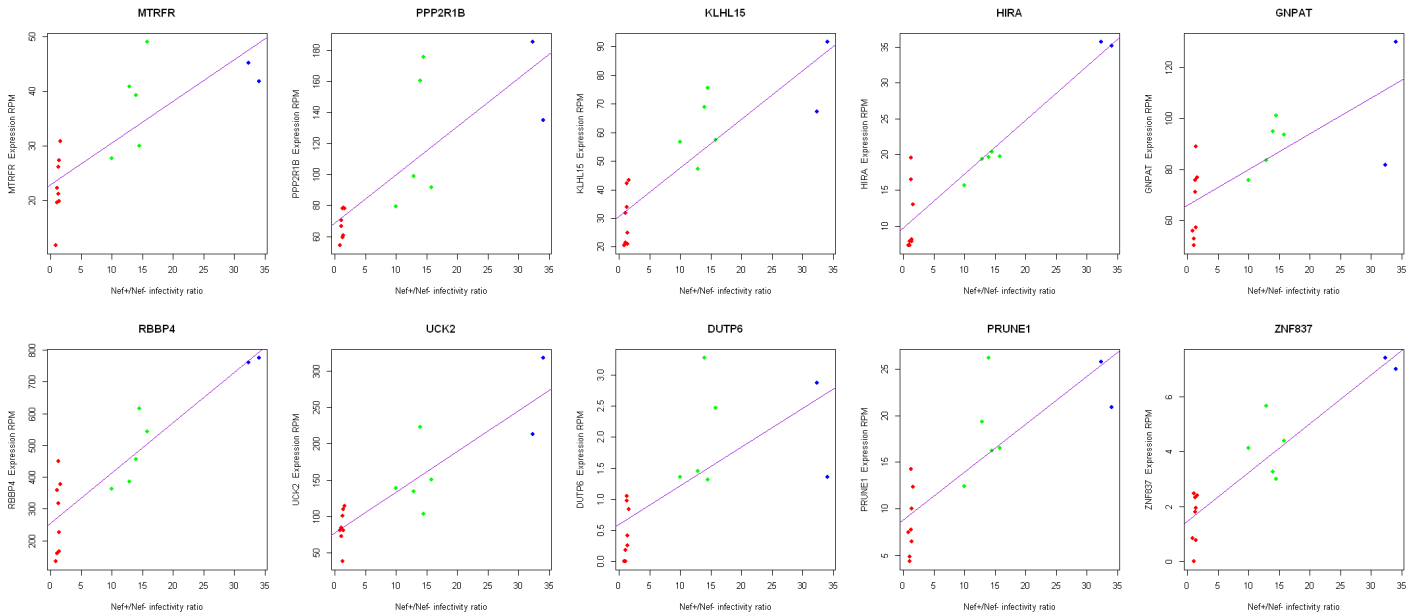
```
        Wilcoxon rank sum exact test

data:  low_Nef_dependent_cell_lines_HIRA_expression and high_Nef_dependent_cell_lines_HIRA_expression
W = 3, p-value = 0.002331
alternative hypothesis: true location shift is less than 0
```

Plotting top positively correlated genes and comparing them with each other

In [34]:
```
correlation_plot <- function(rowdata){
gene_expression_rpm <- as.numeric(unlist(rowdata[2:16]))
sorted_infect_ratio <- arrange(infect_data,V2)$V2
sorted_infect_color <- arrange(infect_data,V2)$V4
plot(sorted_infect_ratio, gene_expression_rpm, pch=16, col=sorted_infect_color, main=rowdata$gene_names, xlab="Nef+/Nef- infectivity ratio", yl
abline(lm(gene_expression_rpm~sorted_infect_ratio), col="darkviolet")
}
```

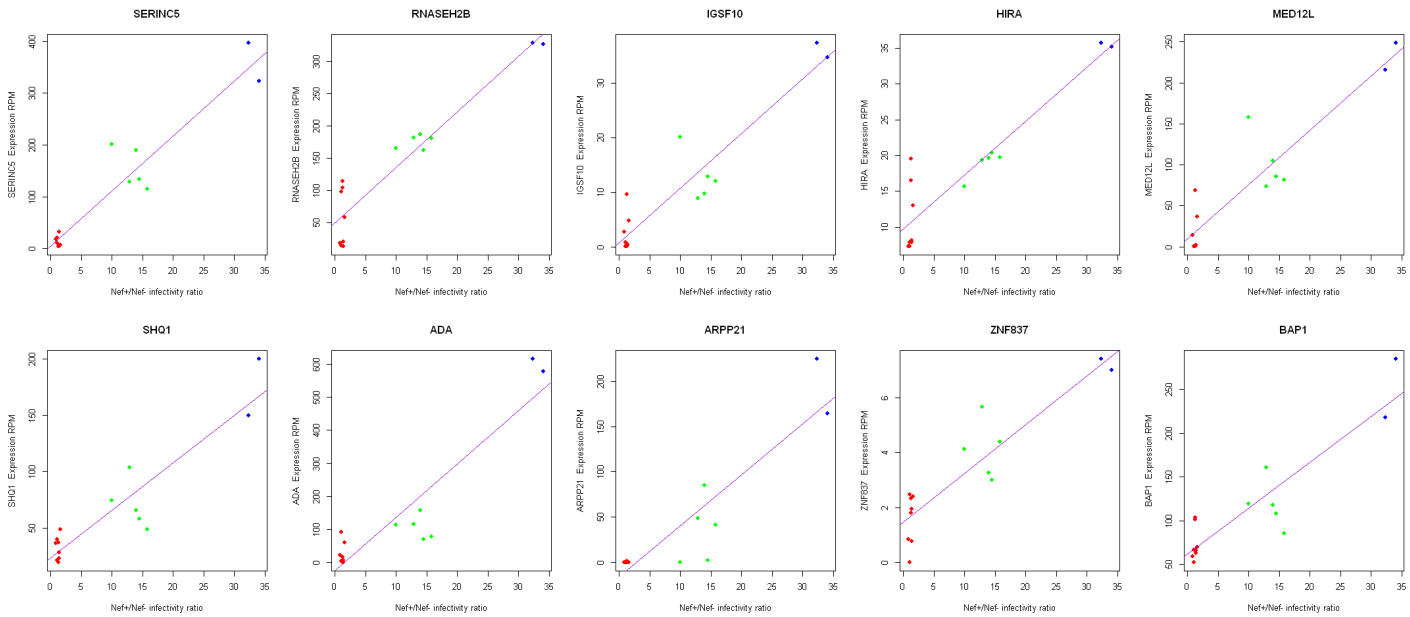
Top 10 according to Spearman correlation

In [35]:
```
top_positively_correlated <- c(1:10)

options(repr.plot.width=18, repr.plot.height=8)
# positively correlated vs infectivity ratio
par(mfrow=c(2,5))
for (i in top_positively_correlated){
  suppressWarnings(correlation_plot(spearman_result_df_sorted[i,]))
}
```



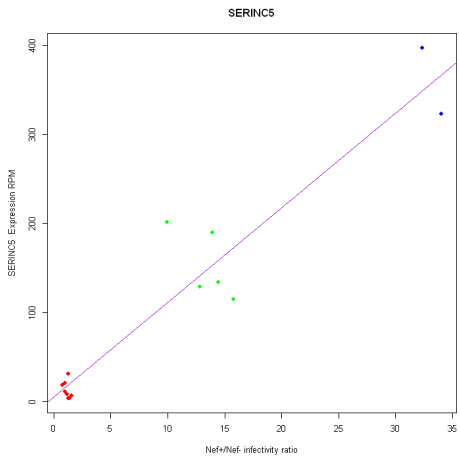
Top 10 according to Pearson correlation

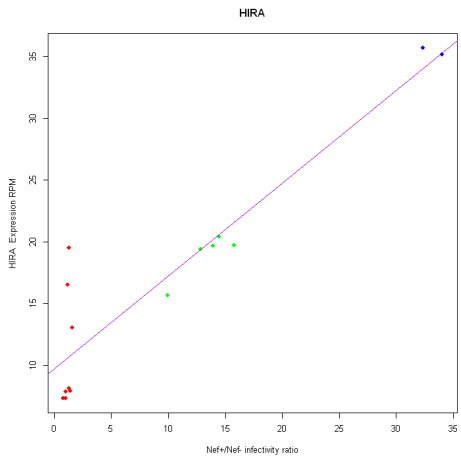
In [36]:
```
top_positively_correlated <- c(1:10)

options(repr.plot.width=18, repr.plot.height=8)
# positively correlated vs infectivity ratio
par(mfrow=c(2,5))
for (i in top_positively_correlated){
  suppressWarnings(correlation_plot(pearson_result_df_sorted[i,]))
}
```

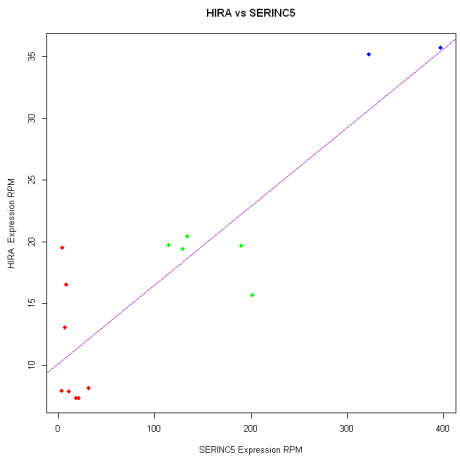


Comparing SERINC5 with HIRA

In [37]:
```
options(repr.plot.width=18, repr.plot.height=6)
par(mfrow=c(1,3))
SERINC5_gene_expression_rpm <- as.numeric(unlist(rpm_df[rpm_df$gene_names=="SERINC5",][2:16]))
HIRA_gene_expression_rpm <- as.numeric(unlist(rpm_df[rpm_df$gene_names=="HIRA",][2:16]))
sorted_infect_ratio <- arrange(infect_data,V2)$V2
sorted_infect_color <- arrange(infect_data,V2)$V4
plot(sorted_infect_ratio, SERINC5_gene_expression_rpm, pch=16, col=sorted_infect_color, main="SERINC5", xlab="Nef+/Nef- infectivity ratio", yla
abline(lm(SERINC5_gene_expression_rpm~sorted_infect_ratio), col="darkviolet")
plot(sorted_infect_ratio, HIRA_gene_expression_rpm, pch=16, col=sorted_infect_color, main="HIRA", xlab="Nef+/Nef- infectivity ratio", ylab=past
abline(lm(HIRA_gene_expression_rpm~sorted_infect_ratio), col="darkviolet")
plot(SERINC5_gene_expression_rpm, HIRA_gene_expression_rpm, pch=16, col=sorted_infect_color, main="HIRA vs SERINC5", xlab="SERINC5 Expression R
abline(lm(HIRA_gene_expression_rpm~SERINC5_gene_expression_rpm), col="darkviolet")
```







HIRA regulating other chromatin remodelling factors

In [38]:
```
HIRA_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="HIRA",][,c(-1,-17)]))

find_cor_hira <- function(onerow, methodname){
sorted_infect_ratio <- arrange(infect_data,V2)$V2
cor_value <- suppressWarnings(cor.test(HIRA_expression,as.numeric(as.vector(unlist(onerow))[c(-1,-17)]), method=methodname))
return(as.numeric(cor_value$estimate))
}

result_df <- rpm_df
result_df$cor_with_infect_ratio <- as.vector(apply(rpm_df,1,find_cor_hira, methodname="spearman"))
head(arrange(result_df, desc(cor_with_infect_ratio)),10)
```

A data.frame: 10 × 18

	ID	HT1080	CEM X 174	IMR90	MT4	DAUDI	WI38	C8166	RAJI	HSB2	CEM SS	CEM A 301	RAMOS
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	ENSG00000100084	7.2954585	7.835726	7.3241827	16.535635	19.53008	8.1183272	7.920350	13.0224038	15.64214	19.397173	19.653380	20.398796
2	ENSG00000162521	134.2644957	358.876232	159.5862124	318.441909	450.98647	166.2961600	226.295728	377.7614901	362.47658	385.114706	455.303301	614.963717
3	ENSG00000082196	0.1402973	1.175359	0.4416593	2.843231	5.80624	0.6477389	1.645787	0.7265719	3.45932	4.041078	6.551127	4.559731
4	ENSG00000196323	46.7189938	101.864433	42.1416543	131.087929	248.71819	56.8714731	52.048017	105.0734726	137.92158	181.444389	275.147318	207.587752
5	ENSG00000153147	170.4611936	182.964192	137.8344939	228.730479	255.05227	183.5260141	212.615123	222.0515461	247.04057	274.065891	429.098794	366.818358
6	ENSG00000163348	22.3072673	25.857894	14.0594864	26.262479	39.37686	23.3185994	17.897935	28.1686331	31.28429	28.125901	29.480070	41.037579
7	ENSG00000038358	36.4772925	46.230781	45.0124395	72.128290	63.65750	59.3328808	57.293964	69.1919995	66.17830	65.950388	78.613520	76.315497
8	ENSG00000106809	1.8238646	5.485008	1.5826123	38.682910	23.43609	2.0295818	5.554532	7.4892794	13.83728	13.739664	32.755633	41.277565
9	ENSG00000178502	15.1521061	17.238596	18.0344198	18.780291	34.73187	15.7184633	21.498094	22.0766073	23.08720	33.379302	36.031196	34.437968
10	ENSG00000196247	9.2596204	67.779026	10.1581630	129.067738	220.00370	15.7184633	40.630369	82.6056343	127.84444	181.444389	180.155982	398.736475

In [39]:
```
options(repr.plot.width=18, repr.plot.height=4)
par(mfrow=c(1,5))

RBBP4_gene_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="RBBP4",][,c(-1,-17)]))
ZBTB44_gene_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="ZBTB44",][,c(-1,-17)]))
SMARCA5_gene_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="SMARCA5",][,c(-1,-17)]))
PYGO2_gene_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="PYGO2",][,c(-1,-17)]))
EDC4_gene_expression <- as.vector(unlist(rpm_df[rpm_df$gene_names=="EDC4",][,c(-1,-17)]))

plot(RBBP4_gene_expression, HIRA_gene_expression_rpm, pch=16, col=sorted_infect_color, main="HIRA vs RBBP4", xlab="RBBP4 Expression RPM", ylab=
abline(lm(HIRA_gene_expression_rpm~RBBP4_gene_expression), col="darkviolet")
plot(ZBTB44_gene_expression, HIRA_gene_expression_rpm, pch=16, col=sorted_infect_color, main="HIRA vs ZBTB44", xlab="ZBTB44 Expression RPM", yl
abline(lm(HIRA_gene_expression_rpm~ZBTB44_gene_expression), col="darkviolet")
plot(SMARCA5_gene_expression, HIRA_gene_expression_rpm, pch=16, col=sorted_infect_color, main="HIRA vs SMARCA5", xlab="SMARCA5 Expression RPM",
abline(lm(HIRA_gene_expression_rpm~SMARCA5_gene_expression), col="darkviolet")
plot(PYGO2_gene_expression, HIRA_gene_expression_rpm, pch=16, col=sorted_infect_color, main="HIRA vs PYGO2", xlab="PYGO2 Expression RPM", ylab=
abline(lm(HIRA_gene_expression_rpm~PYGO2_gene_expression), col="darkviolet")
plot(EDC4_gene_expression, HIRA_gene_expression_rpm, pch=16, col=sorted_infect_color, main="HIRA vs EDC4", xlab="EDC4 Expression RPM", ylab=pas
abline(lm(HIRA_gene_expression_rpm~EDC4_gene_expression), col="darkviolet")
```