Top 23 connectivity analysis output

Tomas Bjorklund

Mon Nov 2 23:47:06 2020

This script presents top 23 candidates as connectivity graph.

```
suppressPackageStartupMessages(library(knitr))
```

Selection of relevant samples

```
select.samples <- readRDS("data/allSamplesDataTable.RDS")
select.samples <- select.samples[-grep("4wks|mRNA_3cpc_pNeuron_RNA", select.samples$Group),
    ]

select.samples.merge <- data.table::copy(select.samples)
select.samples.merge$Lib <- "30cpc"
select.samples.merge$Lib[grep("3cpc", select.samples.merge$Group)] <- "3cpc"
select.samples.merge[, `:=`(Group, gsub("_30cpc|_3cpc", "", Group))]

setkey(select.samples.merge, Group)
select.samples.merge.binPos <- select.samples.merge[c("mRNA_Str", "mRNA_Th",
    "mRNA_Ctx", "mRNA_SN")]
```

Summarize data over each aa

```
setorder(select.samples.merge.binPos, Group, GeneName, start, width)
winWidth = 1
windowTable <- select.samples.merge.binPos[, c("GeneName", "start", "width"),
    with = FALSE]
windowTable <- unique(windowTable, by = c("GeneName", "start", "width"))
windowTable <- windowTable[, seq(start, (start + width - winWidth)), by = c("GeneName",
    "start", "width")]
setnames(windowTable, "V1", "winStart")
windowTable[, `:=`(winEnd, winStart + winWidth - 1)]
setkeyv(windowTable, c("GeneName", "start", "width"))
setkeyv(select.samples.merge.binPos, c("GeneName", "start", "width"))
select.samples.windowBin <- select.samples.merge.binPos[windowTable, nomatch = 0,
    allow.cartesian = TRUE]

setkeyv(select.samples.windowBin, c("Group", "GeneName", "winStart", "winEnd"))
select.samples.windowBin.out <- select.samples.windowBin[, list(Overlaps = .N,
    BCcount = length(table(strsplit(paste(t(BC), collapse = ","), ","))), NormCount = mean(log2(RNAcount +
        1)), AnimalCount = length(table(strsplit(paste(t(Animals), collapse = ","),
        ","))), LUTnrs = paste(unique(names(table(strsplit(paste(t(LUTnrs),
        collapse = ","), ",")))), collapse = ","), mainStruct = paste(unique(structure),
        collapse = ","), mainLibs = paste(unique(Lib), collapse = ","), libCount = length(unique(Lib)),
    mismatches = median(mismatches)), by = c("Group", "GeneName", "winStart",
    "winEnd", "seqlength")]
```

Identify local maxima for each gene and sample

```
select.samples.windowBin <- data.table::copy(select.samples.windowBin.out)
select.samples.windowBin[, `:=`(Score, BCcount + AnimalCount + libCount)]
setorder(select.samples.windowBin, Group, -Score, -AnimalCount, -BCcount, -libCount,
    -NormCount, GeneName, winStart)
select.samples.windowBin[, `:=`(Rank, seq(.N)), by = c("Group")]
setorder(select.samples.windowBin, Group, GeneName, winStart, -Score)


windowTable <- select.samples.windowBin[, c("Group", "GeneName", "seqlength"),
    with = FALSE]
windowTable <- unique(windowTable, by = c("Group", "GeneName", "seqlength"))
windowTable <- windowTable[, seq(seqlength), by = c("Group", "GeneName")]
setnames(windowTable, "V1", "winStart")
setkeyv(windowTable, c("Group", "GeneName", "winStart"))
setkeyv(select.samples.windowBin, c("Group", "GeneName", "winStart"))
select.samples.windowBin <- select.samples.windowBin[windowTable]
select.samples.windowBin$Score[is.na(select.samples.windowBin$Score)] <- 0

select.samples.windowBin$LUTnrs[is.na(select.samples.windowBin$LUTnrs)] <- seq(length(which(is.na(select.sar

setorder(select.samples.windowBin, Group, GeneName, -winStart)
select.samples.windowBin.locMax <- data.table::copy(select.samples.windowBin)
select.samples.windowBin.locMax[, `:=`(Peak, as.logical(extract(turnpoints(Score),
    peak = 1, pit = 0))), by = c("Group", "GeneName")]  #,proba=0.0001

setorder(select.samples.windowBin.locMax, Group, GeneName, winStart)
select.samples.windowBin.locMax <- select.samples.windowBin.locMax[select.samples.windowBin.locMax$Peak,
    ]
select.samples.windowBin.locMax[, `:=`(Peak, NULL)]

setorder(select.samples.windowBin.locMax, Group, Rank, -Score, GeneName, winStart)
select.samples.windowBin.locMax <- unique(select.samples.windowBin.locMax, by = c("Group",
    "LUTnrs"))
```

Make binning for 22aa fragment length

```
# Connect highest scoring fragments

setorder(select.samples.windowBin.locMax, Group, -Score, -AnimalCount, -BCcount,
    -libCount, -NormCount, GeneName, winStart)
setkey(select.samples.windowBin.locMax, Group)
select.samples.windowBin.locMax[, `:=`(Rank2, seq(.N)), by = c("Group")]


select.samples.topSelect <- select.samples.windowBin.locMax[, head(.SD, 35),
    by = Group]


setorder(select.samples.topSelect, Group, GeneName, winStart)
winWidth = 21
windowTable <- select.samples.topSelect[, c("GeneName", "winStart"), with = FALSE]
windowTable <- unique(windowTable, by = c("GeneName", "winStart"))
```

```
windowTable <- windowTable[, seq((winStart - winWidth), (winStart)), by = c("GeneName",
    "winStart")]
setnames(windowTable, "V1", "binBaseStart")
windowTable <- windowTable[binBaseStart > 0, ]
windowTable[, `:=`(binBaseEnd, binBaseStart + winWidth)]
scoreSelect <- select.samples.topSelect[, c("Group", "GeneName", "winStart",
    "Score"), with = FALSE]
setkeyv(windowTable, c("GeneName", "winStart"))
setkeyv(scoreSelect, c("GeneName", "winStart"))
scoreSelect <- scoreSelect[windowTable, allow.cartesian = TRUE]

scoreSelect[, `:=`(mCount, length(unique(Group))), by = c("GeneName", "binBaseStart")]
scoreSelect[, `:=`(oCount, .N), by = c("GeneName", "binBaseStart")]
scoreSelect[, `:=`(tCount, mCount + oCount)]
scoreSelect[, `:=`(offset, abs(binBaseStart + 6 - mean(winStart))), c("GeneName",
    "binBaseStart")]
# scoreSelect <- scoreSelect[mCount>oCount,]
setorder(scoreSelect, GeneName, winStart, -tCount, binBaseStart, -offset)

scoreSelect <- scoreSelect[, head(.SD, 1), by = c("GeneName", "winStart")]

scoreSelect <- scoreSelect[, c("GeneName", "winStart", "binBaseStart", "binBaseEnd"),
    with = FALSE]
setkeyv(scoreSelect, c("GeneName", "winStart"))
setkeyv(select.samples.windowBin.locMax, c("GeneName", "winStart"))


select.samples.windowBin.locMax.bin <- scoreSelect[select.samples.windowBin.locMax,
    nomatch = 0]
setorder(select.samples.windowBin.locMax.bin, GeneName, -binBaseStart, -Score,
    Group)

select.samples.windowBin.locMerge <- select.samples.windowBin.locMax.bin[, head(.SD,
    1), by = c("GeneName", "binBaseStart", "binBaseEnd", "Group")]
select.samples.windowBin.locMerge <- unique(select.samples.windowBin.locMerge,
    by = c("Group", "GeneName", "LUTnrs"))
```

Selection of top twenty fragments per sample

```
setorder(select.samples.windowBin.locMerge, Group, -Score, -AnimalCount, -BCcount,
    -libCount, -NormCount, GeneName, binBaseStart)
setkey(select.samples.windowBin.locMerge, Group)

select.samples.topTwenty <- select.samples.windowBin.locMerge[, head(.SD, 25),
    by = Group]
select.samples.topTwenty <- select.samples.topTwenty[, c("GeneName", "binBaseStart",
    "binBaseEnd"), with = FALSE]
select.samples.topTwenty <- unique(select.samples.topTwenty, by = c("GeneName",
    "binBaseStart"))

setorder(select.samples.windowBin, Group, -Score, -AnimalCount, -BCcount, -libCount,
    -NormCount, GeneName, winStart)
select.samples.windowBin[, `:=`(Rank, seq(.N)), by = c("Group")]
setkeyv(scoreSelect, c("GeneName", "winStart"))
setkeyv(select.samples.windowBin, c("GeneName", "winStart"))
```

```
select.samples.windowBin <- scoreSelect[select.samples.windowBin, nomatch = 0]
setorder(select.samples.windowBin, Group, GeneName, binBaseStart, -Score)
select.samples.windowBin.Bin <- unique(select.samples.windowBin, by = c("Group",
    "GeneName", "binBaseStart"))

setkeyv(select.samples.topTwenty, c("GeneName", "binBaseStart"))
setkeyv(select.samples.windowBin, c("GeneName", "binBaseStart"))
select.samples.windowBin.allTop <- select.samples.windowBin[select.samples.topTwenty,
    nomatch = 0]
select.samples.windowBin.allTop <- unique(select.samples.windowBin.allTop, by = c("Group",
    "LUTnrs"))
setorder(select.samples.windowBin.allTop, Group, GeneName, binBaseStart, -Score)
select.samples.windowBin.allTop <- unique(select.samples.windowBin.allTop, by = c("Group",
    "GeneName", "binBaseStart"))

select.samples.windowBin.allTop[, `:=`(GeneAA, paste(GeneName, " [", binBaseStart +
    floor(winWidth/2), "]", sep = "")), by = Group]
setorder(select.samples.windowBin.allTop, Group, Rank)
```

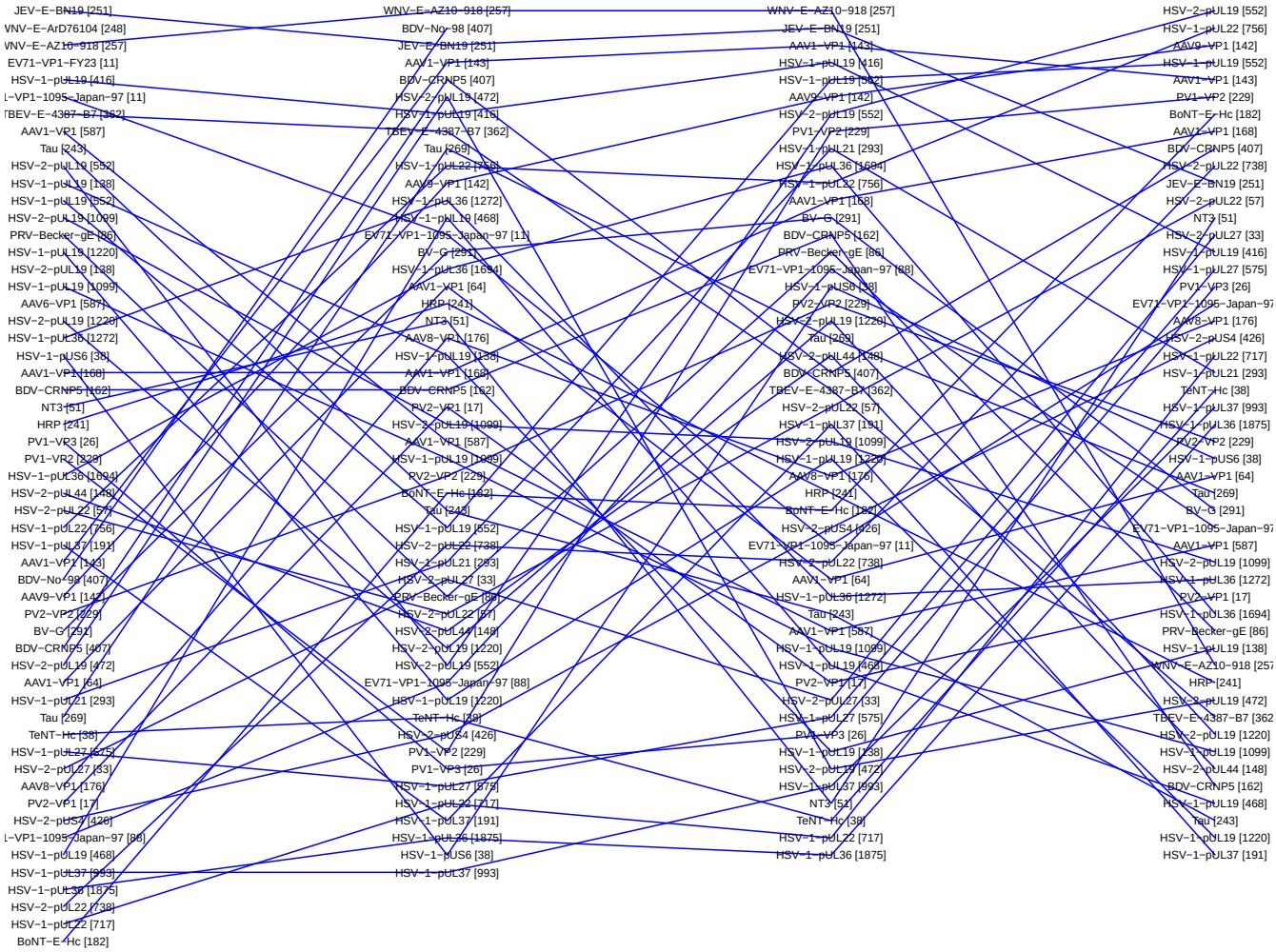
Plotting ranked order

```
setkey(select.samples.windowBin.allTop, Group)

v1 <- select.samples.windowBin.allTop["mRNA_Str"]$GeneAA
v2 <- select.samples.windowBin.allTop["mRNA_Th"]$GeneAA
v3 <- select.samples.windowBin.allTop["mRNA_Ctx"]$GeneAA
v4 <- select.samples.windowBin.allTop["mRNA_SN"]$GeneAA


o <- 0.05
DF <- data.table(x = c(rep(1, length(v1)), rep(2, length(v2)), rep(3, length(v3)),
    rep(4, length(v4))), x1 = c(rep(1, length(v1)), rep(2, length(v2)), rep(3,
    length(v3)), rep(4, length(v4))), y = c(rev(seq_along(v1)), rev(seq_along(v2)),
    rev(seq_along(v3)), rev(seq_along(v4))), g = c(v1, v2, v3, v4))
DF[, `:=`(groupMax, max(y)), by = x]
allMax <- max(DF$y)
DF[, `:=`(y, y - groupMax + allMax)]


ggplot(DF, aes(x = x, y = y, group = g, label = g)) + geom_path(aes(x = x1),
    size = 0.3, color = "blue") + geom_text(size = 1.8) + theme_minimal() +
    theme(axis.title = element_blank(), axis.text = element_blank(), axis.ticks = element_blank(),
        panel.grid = element_blank())
```


JEV−E−BN19 [251]
VNV−E−ArD76104 [248]
VNV−E−AZ10−918 [257]
EV71−VP1−FY23 [11]
HSV−1−pUL19 [416]
L−VP1−1095−Japan−97 [11]
TBEV−E−4387−B7 [362]
AAV1−VP1 [587]
Tau [243]
HSV−2−pUL19 [552]
HSV−1−pUL19 [138]
HSV−1−pUL19 [552]
HSV−2−pUL19 [1099]
PRV−Becker−gE [86]
HSV−1−pUL19 [1220]
HSV−2−pUL19 [138]
HSV−1−pUL19 [1099]
AAV6−VP1 [587]
HSV−2−pUL19 [1220]
HSV−1−pUL36 [1272]
HSV−1−pUS6 [38]
AAV1−VP1 [168]
BDV−CRNP5 [162]
NT3 [51]
HRP [241]
PV1−VP3 [26]
PV1−VP2 [229]
HSV−1−pUL36 [1694]
HSV−2−pUL44 [148]
HSV−2−pUL22 [57]
HSV−1−pUL22 [756]
HSV−1−pUL37 [191]
AAV1−VP1 [143]
BDV−No−98 [407]
AAV9−VP1 [142]
PV2−VP2 [229]
BV−G [291]
BDV−CRNP5 [407]
HSV−2−pUL19 [472]
AAV1−VP1 [64]
HSV−1−pUL21 [293]
Tau [269]
TeNT−Hc [38]
HSV−1−pUL27 [575]
HSV−2−pUL27 [33]
AAV8−VP1 [176]
PV2−VP1 [17]
HSV−2−pUS4 [426]
L−VP1−1095−Japan−97 [88]
HSV−1−pUL19 [468]
HSV−1−pUL37 [993]
HSV−1−pUL36 [1875]
HSV−2−pUL22 [738]
HSV−1−pUL22 [717]
BoNT−E−Hc [182]
WNV−E−AZ10−918 [257]
BDV−No−98 [407]
JEV−E−BN19 [251]
AAV1−VP1 [143]
BDV−CRNP5 [407]
HSV−2−pUL19 [472]
HSV−1−pUL19 [416]
TBEV−E−4387−B7 [362]
Tau [269]
HSV−1−pUL22 [756]
AAV9−VP1 [142]
HSV−1−pUL36 [1272]
HSV−1−pUL19 [468]
EV71−VP1−1095−Japan−97 [11]
BV−G [291]
HSV−1−pUL36 [1694]
AAV1−VP1 [64]
HRP [241]
NT3 [51]
AAV8−VP1 [176]
HSV−1−pUL19 [138]
AAV1−VP1 [168]
BDV−CRNP5 [162]
PV2−VP1 [17]
HSV−2−pUL19 [1099]
AAV1−VP1 [587]
HSV−1−pUL19 [1099]
PV2−VP2 [229]
BoNT−E−Hc [182]
Tau [243]
HSV−1−pUL19 [552]
HSV−2−pUL22 [738]
HSV−1−pUL21 [293]
HSV−2−pUL27 [33]
PRV−Becker−gE [86]
HSV−2−pUL22 [57]
HSV−2−pUL44 [148]
HSV−2−pUL19 [1220]
HSV−2−pUL19 [552]
EV71−VP1−1095−Japan−97 [88]
HSV−1−pUL19 [1220]
TeNT−Hc [38]
HSV−2−pUS4 [426]
PV1−VP2 [229]
PV1−VP3 [26]
HSV−1−pUL27 [575]
HSV−1−pUL22 [717]
HSV−1−pUL37 [191]
HSV−1−pUL36 [1875]
HSV−1−pUS6 [38]
HSV−1−pUL37 [993]
WNV−E−AZ10−918 [257]
JEV−E−BN19 [251]
AAV1−VP1 [143]
HSV−1−pUL19 [416]
HSV−1−pUL19 [552]
AAV9−VP1 [142]
HSV−2−pUL19 [552]
PV1−VP2 [229]
HSV−1−pUL21 [293]
HSV−1−pUL36 [1694]
HSV−1−pUL22 [756]
AAV1−VP1 [168]
BV−G [291]
BDV−CRNP5 [162]
PRV−Becker−gE [86]
EV71−VP1−1095−Japan−97 [88]
HSV−1−pUS6 [38]
PV2−VP2 [229]
HSV−2−pUL19 [1220]
Tau [269]
HSV−2−pUL44 [148]
BDV−CRNP5 [407]
TBEV−E−4387−B7 [362]
HSV−2−pUL22 [57]
HSV−1−pUL37 [191]
HSV−2−pUL19 [1099]
HSV−1−pUL19 [1220]
AAV8−VP1 [176]
HRP [241]
BoNT−E−Hc [182]
HSV−2−pUS4 [426]
EV71−VP1−1095−Japan−97 [11]
HSV−2−pUL22 [738]
AAV1−VP1 [64]
HSV−1−pUL36 [1272]
Tau [243]
AAV1−VP1 [587]
HSV−1−pUL19 [1099]
HSV−1−pUL19 [468]
PV2−VP1 [17]
HSV−2−pUL27 [33]
HSV−1−pUL27 [575]
PV1−VP3 [26]
HSV−1−pUL19 [138]
HSV−2−pUL19 [472]
HSV−1−pUL37 [993]
NT3 [51]
TeNT−Hc [38]
HSV−1−pUL22 [717]
HSV−1−pUL36 [1875]
HSV−2−pUL19 [552]
HSV−1−pUL22 [756]
AAV9−VP1 [142]
HSV−1−pUL19 [552]
AAV1−VP1 [143]
PV1−VP2 [229]
BoNT−E−Hc [182]
AAV1−VP1 [168]
BDV−CRNP5 [407]
HSV−2−pUL22 [738]
JEV−E−BN19 [251]
HSV−2−pUL22 [57]
NT3 [51]
HSV−2−pUL27 [33]
HSV−1−pUL19 [416]
HSV−1−pUL27 [575]
PV1−VP3 [26]
EV71−VP1−1095−Japan−97
AAV8−VP1 [176]
HSV−2−pUS4 [426]
HSV−1−pUL22 [717]
HSV−1−pUL21 [293]
TeNT−Hc [38]
HSV−1−pUL37 [993]
HSV−1−pUL36 [1694]
HSV−1−pUL36 [1099]
PV2−VP2 [229]
HSV−1−pUS6 [38]
AAV1−VP1 [64]
Tau [269]
BV−G [291]
EV71−VP1−1095−Japan−97
AAV1−VP1 [587]
HSV−2−pUL19 [1099]
HSV−1−pUL36 [1272]
PV2−VP1 [17]
HSV−1−pUL36 [1694]
PRV−Becker−gE [86]
HSV−1−pUL19 [138]
WNV−E−AZ10−918 [257]
HRP [241]
HSV−2−pUL19 [472]
TBEV−E−4387−B7 [362]
HSV−2−pUL19 [1220]
HSV−1−pUL19 [1099]
HSV−2−pUL44 [148]
BDV−CRNP5 [162]
HSV−1−pUL19 [468]
Tau [243]
HSV−1−pUL19 [1220]
HSV−1−pUL37 [191]