

Supplementary Figure 2

Contents

Supplementary Figure 2A-D. Examples of linear and radial dendograms and commands to generate them	1
Supplementary Figure 2E-G. Merge motifs using the <i>motifSignature</i> function with different distance cutoffs, and plot merged motifs in various layouts	5
Supplementary Figure 2H-M. Use various color options to visualize different sources of variations in the motif set	9
Supplementary Figure 2N-O. Compare different Column Comparison Metrics (CCM) and alignment methods	17
Supplementary Figure 2P. Different distance resolution between MotIV and MatAlign even when using the same Column Comparison Matrix and alignment method	21

Input Data

```
##load the library
library(motifStack)

##read pcms
pcmpath <- "pcmsDatasetDM"
pcms <- readPCM(pcmpath)
##convert to pfms
pfms<-lapply(pcms,pcm2pfm)
```

Supplementary Figure 2A-D. Examples of linear and radial dendograms and commands to generate them

Plot motifs as a stack

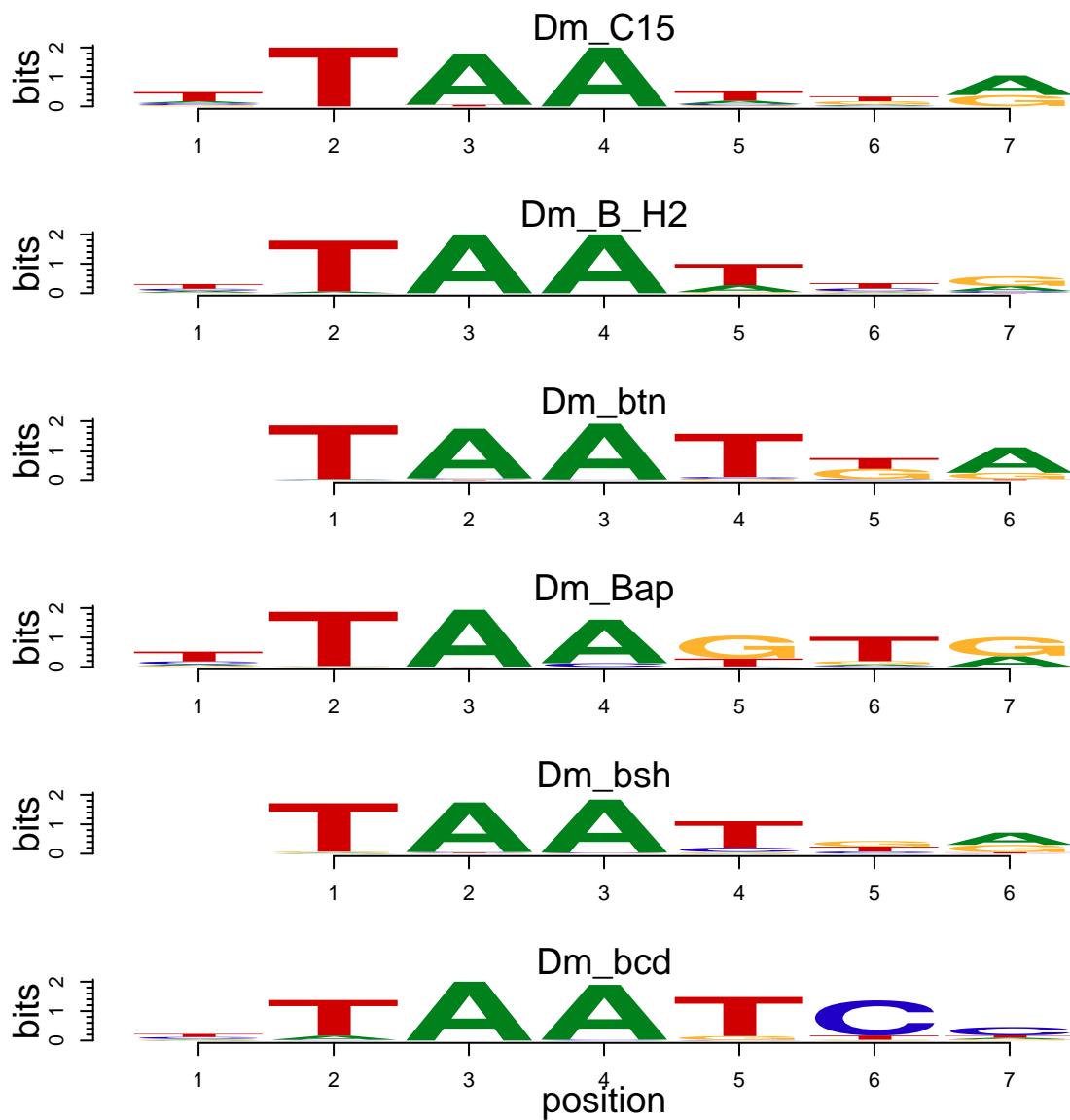
```
subset_pfms <- pfms[10:15]
motifStack(subset_pfms, layout="stack")
```

Plot motifs as a linear tree

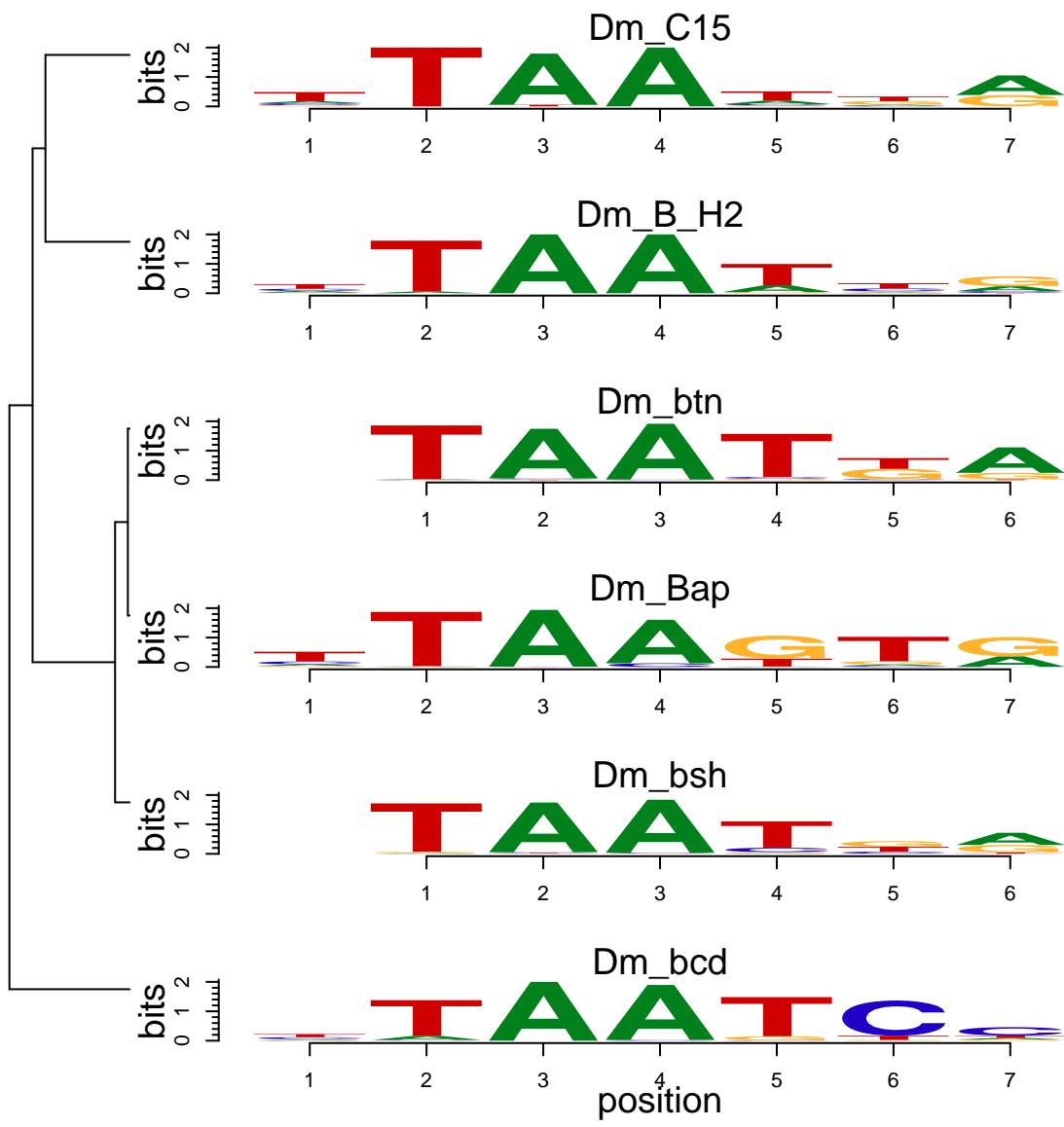
```
##By default, MotIV is used for clustering
motifStack(subset_pfms, layout="tree", trueDist=TRUE)
```

Plot motifs as a linear phylogenetic tree

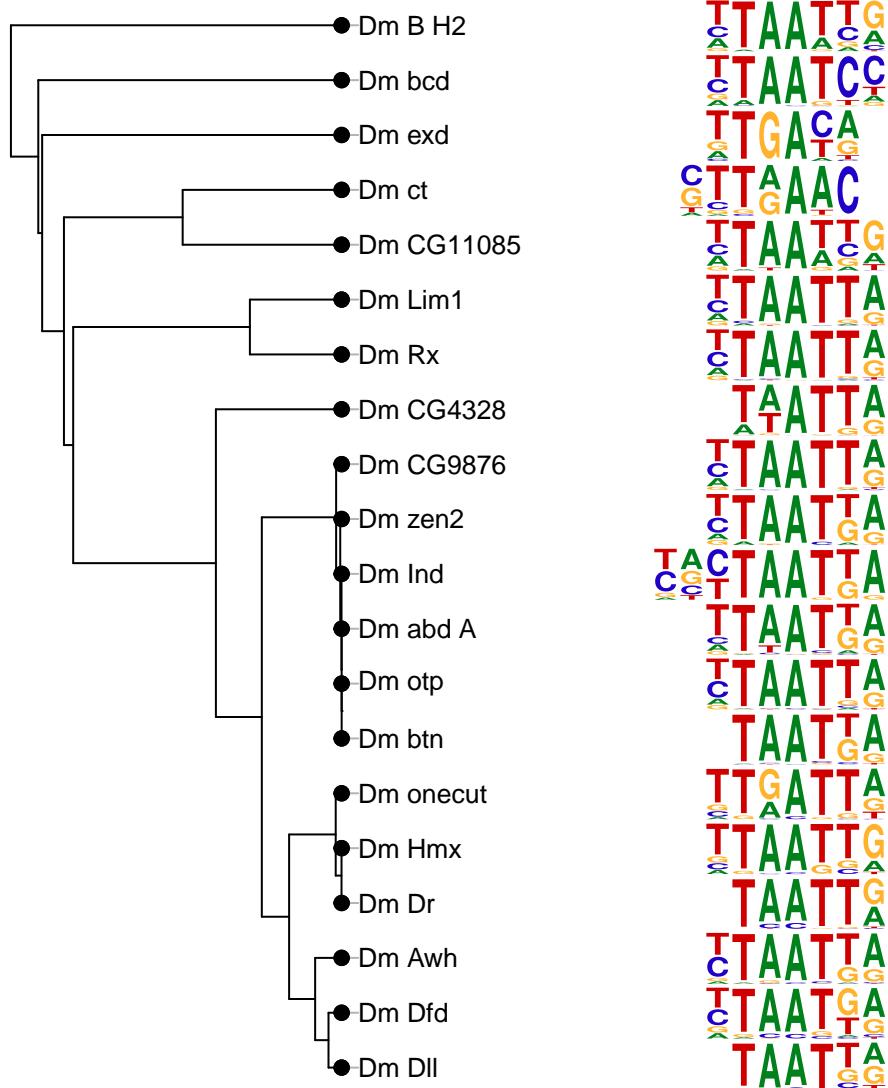
```
##try different style of sequence logo
motifStack(pfms[sample(1:length(pfms), 20)], layout="phylog", clabel.leaves=.8,
f.logo=0.6, ic.scale=FALSE)
```



Supplementary Figure 2A: plotted as a stack of motifs



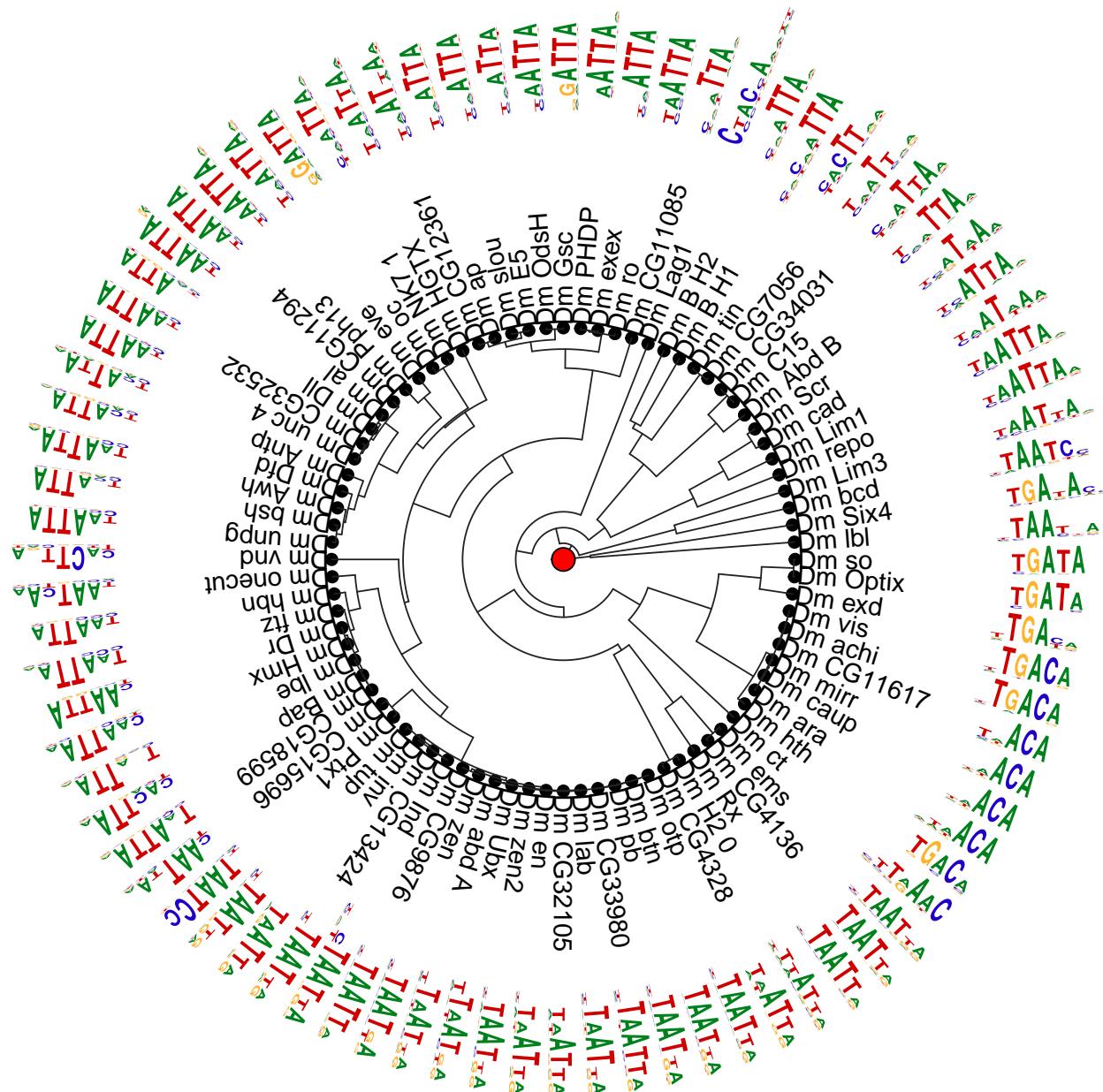
Supplementary Figure 2B: plotted as a linear tree



Supplementary Figure 2C: plotted as a linear phylogenetic tree

Plot motifs as a radial phylogenetic tree

```
motifStack(pfms, layout="radialPhylog")
```



Supplementary Figure 2D: plotted as a radial phylogenetic tree

Supplementary Figure 2E-G. Merge motifs using the *motifSignature* function with different distance cutoffs, and plot merged motifs in various layouts

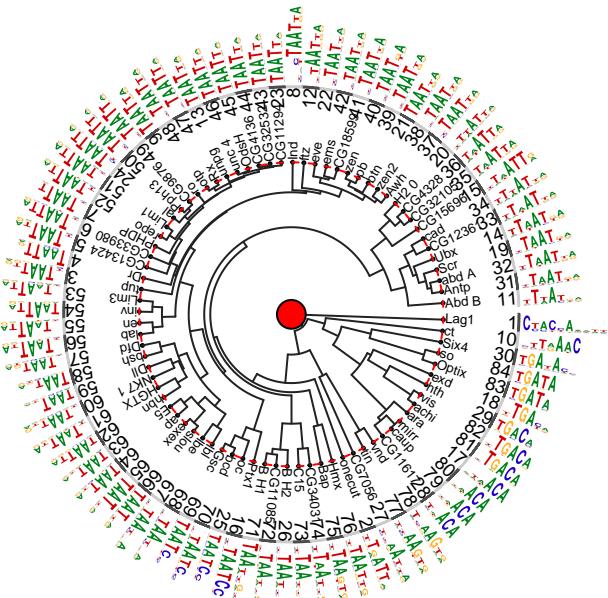
Before motifs being merged, the distances of the motifs need to be calculated using STAMP, MovIV or MatAlign. Here we show different ways to draw the merged motifs and how different distance cutoffs affect the merging results. The red dotted line indicates the distance cutoff used.

Merge motifs using different distance cutoffs

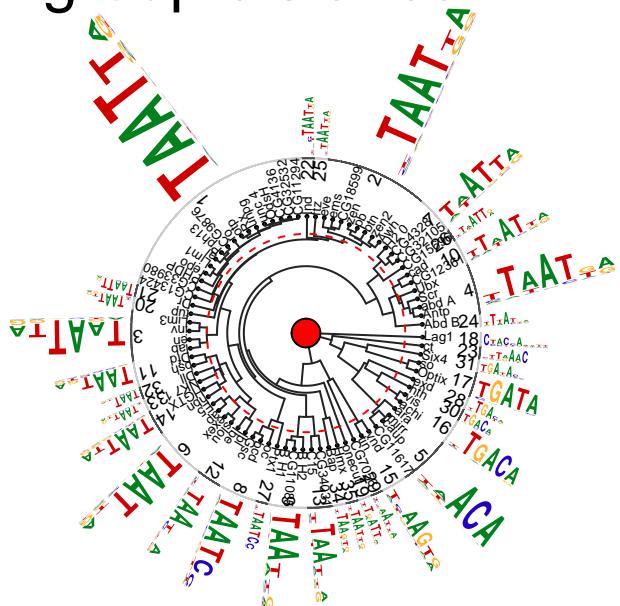
```
source("zzz.R") ## set outpath, matalign_path, neighbor_pah and MatAlign2tree_path
# outpath <- "output"
# matalign_path <- "./app/matalign-v4a"
# neighbor_path <- "./app/neighbor.app/Contents/MacOS/neighbor"
pcmpath <- "pcmsDatasetDM"
pcms <- readPCM(pcmpath)
pfms<-lapply(pcms,pcm2pfm)
system(paste("perl", MatAlign2tree_path, "--in . --pcmpath", pcmpath,
             "--out", outpath,
             "--matalign", matalign_path,
             "--neighbor", neighbor_path,
             "--tree","UPGMA"))
newickstrUPGMA <- readLines(con=file.path(outpath, "NJ.matalign.distMX.nwk"))
phylogUPGMAmatAlign <- newick2phylog(newickstrUPGMA, FALSE)
##get the leaves of phylog to reorder the pfms
matAlignLeaveNames <- names(phylogUPGMAmatAlign$leaves)
this_motifs <- pfms[matAlignLeaveNames]
matAlignLeaveNames <- gsub("^Dm_", "", matAlignLeaveNames)

for(groupDistance in c(0, 1, 2, 3)){
  motifSig <- motifSignature(this_motifs,
                               phylogUPGMAmatAlign,
                               groupDistance=groupDistance,
                               min.freq=1)
  this_sig <- signatures(motifSig)
  ## get color set for signature groups
  this_gpCol <- sigColor(motifSig)
  plotMotifStackWithRadialPhylog(phylog=phylogUPGMAmatAlign,
                                  pfms=this_sig,
                                  col.inner.label.circle=this_gpCol,
                                  inner.label.circle.width=0.02,
                                  labels.leaves=matAlignLeaveNames,
                                  cleaves=.2, circle=1, circle.motif=1.5,
                                  xlabel.leaves=.4, motifScale="logarithmic",
                                  angle=358, plotIndex=TRUE, IndexCex=.6,
                                  groupDistance=groupDistance)
  text(0, 2.3, label=paste("group distance =", groupDistance), cex=2)
}
```

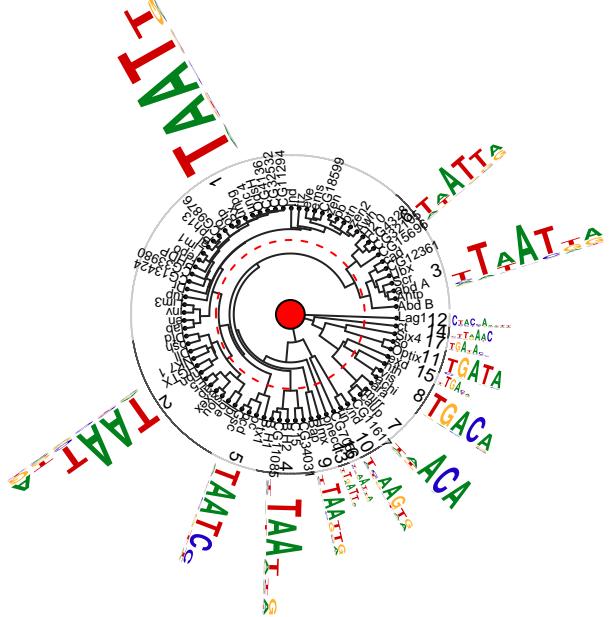
group distance = 0



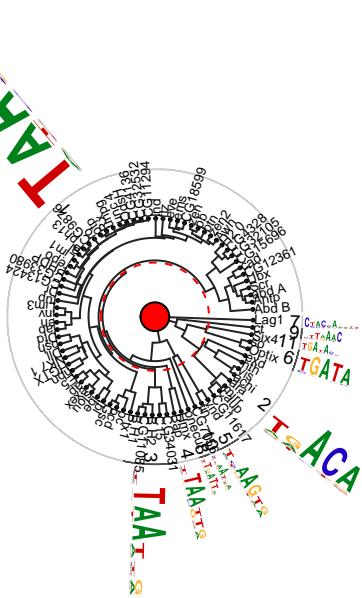
group distance = 1



group distance = 2



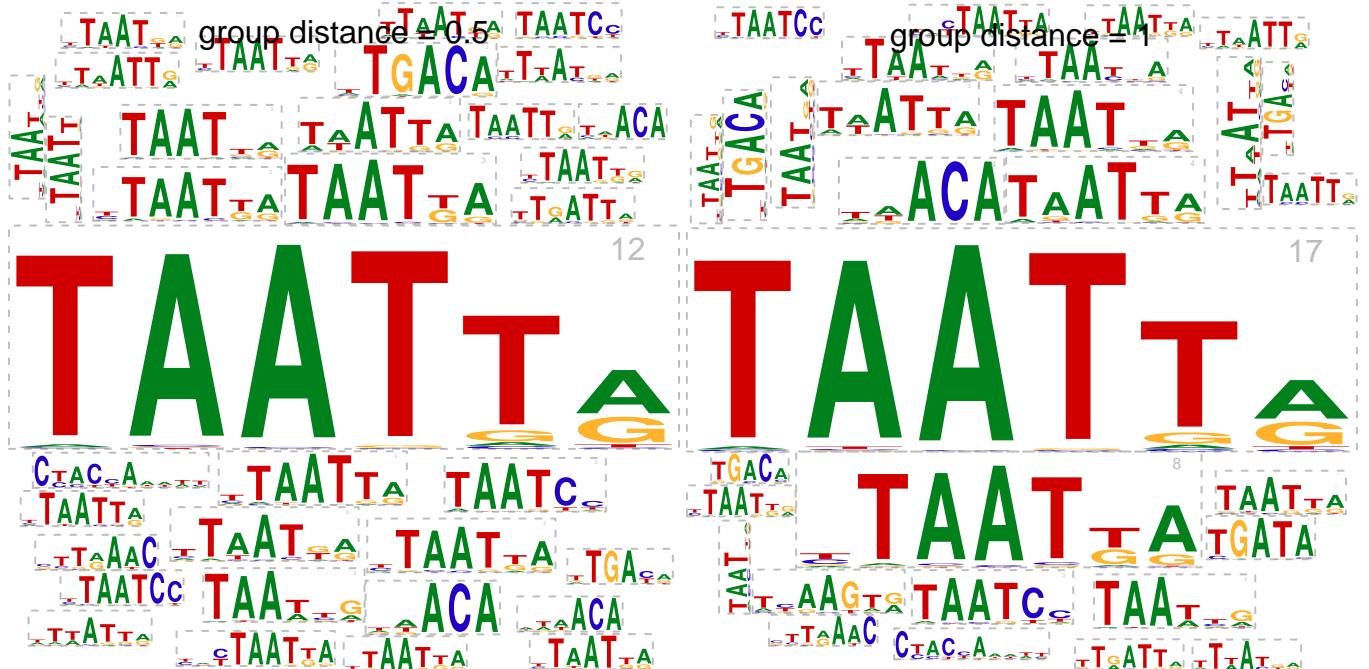
group distance = 3



Plot merged motif as a radial phylogenetic tree

```
## get signature
gpDist <- c(.5, 1)
motifSig <- lapply(gpDist, motifSignature,
  pfms=this_motifs,
  phylog=phylogUPGMAmatAlign,
  min.freq=1)
## motif cloud, cloud style
```

```
for(i in 1:2){  
  motifCloud(motifSig[[i]], layout="cloud", scale=c(9, .75))  
  op <- par(mar = c(0, 0, 0, 0))  
  text(.5, .985, label=paste("group distance =", gpDist[i]))  
  par(op)  
}
```



Plot merged motif as a word cloud

```
## motif cloud, rectangle style
for(i in 1:2){
  motifCloud(motifSig[[i]], layout="rectangles", ic.scale=FALSE)
  op <- par(mar = c(0, 0, 0, 0))
  text(.5, .985, label=paste("group distance =", gpDist[i]))
  par(op)
}
```



Supplementary Figure 2H-M. Use various color options to visualize different sources of variations in the motif set

There are four ways to color a radial phylogenetic tree, i.e., the background of the inner circle, the motif names, the inner label ring and the outer label ring. Here are a few examples on how to use these options to show the data source of the motif, the computational algorithm for generating the motifs, information content (IC) and the motif group.

```

getPairColor <- function(n=10L){
  if(n %% 2 != 0) n <- n+1
  n <- n/2
  n <- rainbow(n)#as.character(t(matrix(rainbow(n=n), ncol=2, byrow=FALSE)))
  n2 <- highlightCol(n, .5)
  as.character(t(cbind(n, n2)))
}

pairColor <- getPairColor(22)

pfmList2matrixList <- function(pfms){
  m <- lapply(pfms, function(.ele) as(.ele, "matrix"))
  names(m) <- unlist(lapply(pfms, function(.ele) .ele@name))
  m
}

getMotIVOut <- function(pfms, cc, align){
  jaspar.scores <-
    MotIV::readDBScores(
      file.path(".", "app", "scores",
                paste("JaspRand_", cc, "_", align, ".scores", sep="")))
  d <- MotIV::motifDistances(pfmList2matrixList(pfms), cc=cc, align=align)
  hc <- MotIV::motifHclust(d, method="average")
  phylog <- hclust2phylog(hc)
  pfms <- pfms[hc$order]
}

```

```

aligned.pfms <- DNAmotifAlignment(pfms)
leaveNames <- names(phylog$leaves)
## data source
dataSource <- factor(grep1("_M", leaveNames))
levels(dataSource) <- c("yellow", "blue") ##c("Uniprobe", "CIS-BP")
dataSource <- as.character(dataSource)
## algorithm
algorithm <-
  factor((!grep1("_M", leaveNames)) + grep1("_bml", leaveNames))
levels(algorithm) <-
  c("blue", "brown", "orange") ##("DREAM5", "Seed-And-Wobble", "BEEML")
algorithm <- as.character(algorithm)
## motifs from same PBM data
motifGroup <- factor(gsub("(.*?)_.*$", "\\\1", leaveNames))
levels.motifGroup <- levels(motifGroup)
levels(motifGroup) <- pairColor[1:length(levels(motifGroup))]
colors.motifGroup <- levels(motifGroup)
motifGroup <- as.character(motifGroup)
return(list(aligned.pfms=aligned.pfms, unaligned.pfms=pfms, phylog=phylog,
           leaveNames=leaveNames,
           dataSource=dataSource,
           algorithm=algorithm,
           motifGroup=motifGroup,
           levels.motifGroup=levels.motifGroup,
           colors.motifGroup=colors.motifGroup))
}

##read pcms
pcmpath <- "pcmsDatasetAlgorithm"
pcms <- readPCM(pcmpath)
##convert to pfms
pfms<-lapply(pcms,pcm2pfm)

```

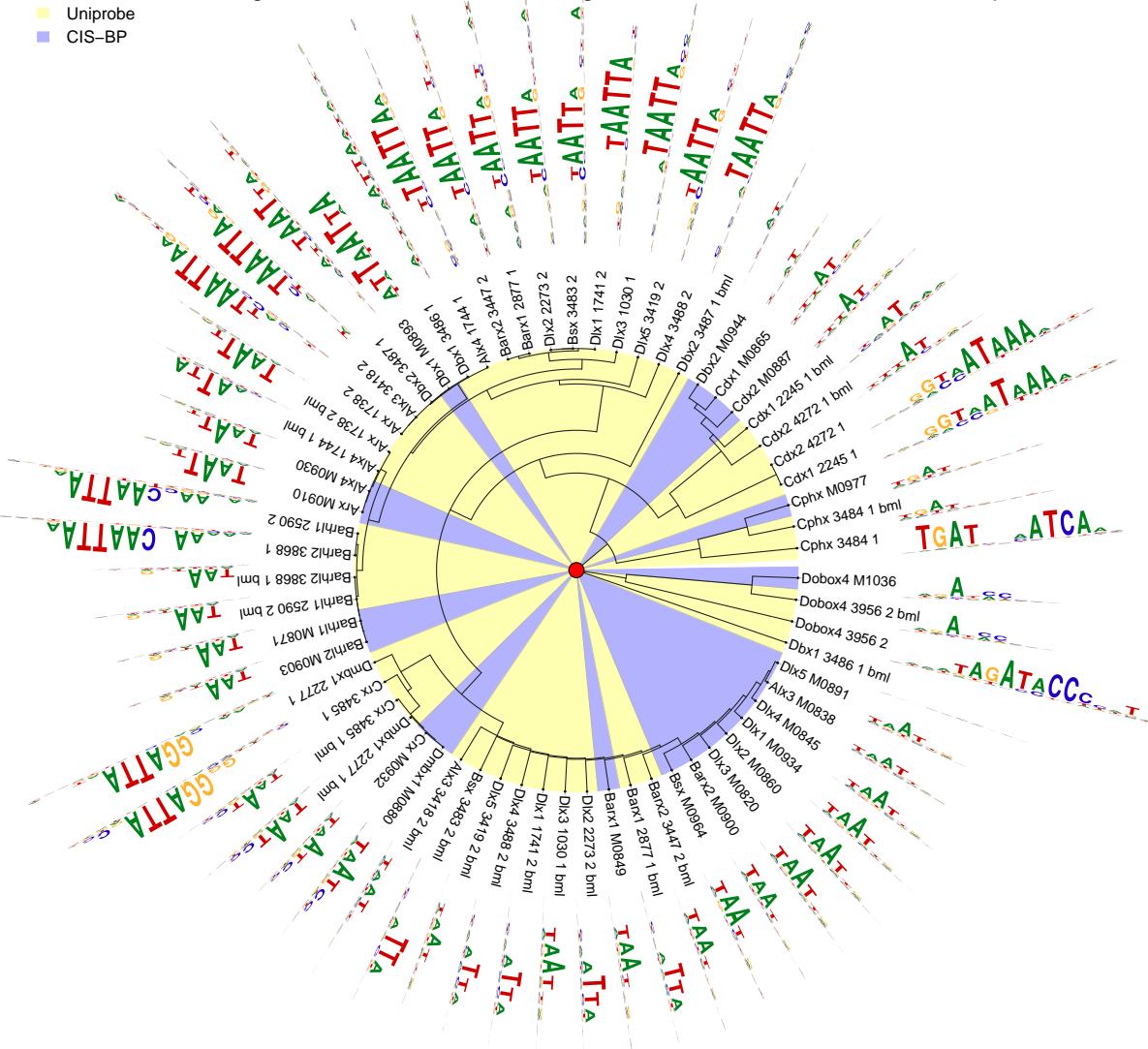
Use the color of the background of inner circle to distinguish different data sources with the parameter col.bg

```

motIVout <- getMotIVOut(pfms, "PCC", "SWU")
attach(motIVout)
plotMotifStackWithRadialPhylog(phylog=phylog, pfms=unaligned.pfms,
                                labels.leaves=leaveNames,
                                col.bg=dataSource, col.bg.alpha=.3,
                                cleaves=.2, circle=1.1, circle.motif=1.6,
                                clabel.leaves=.8, angle=358)
legend(-2.3, 2.4, legend=c("Uniprobe", "CIS-BP"),
       fill= highlightCol(c("yellow", "blue"), alpha=.3),
       border="white", lty=NULL, bty = "n", cex=1)
text(0, 2.4, "Use the color of the background of inner circle to distinguish different data sources with")

```

Use the color of the background of inner circle to distinguish different data sources with the parameter col.bg



Supplementary Figure 2H: color setting, tree background

Use the color of motif names to distinguish the computational algorithms used with the parameter col.leaves

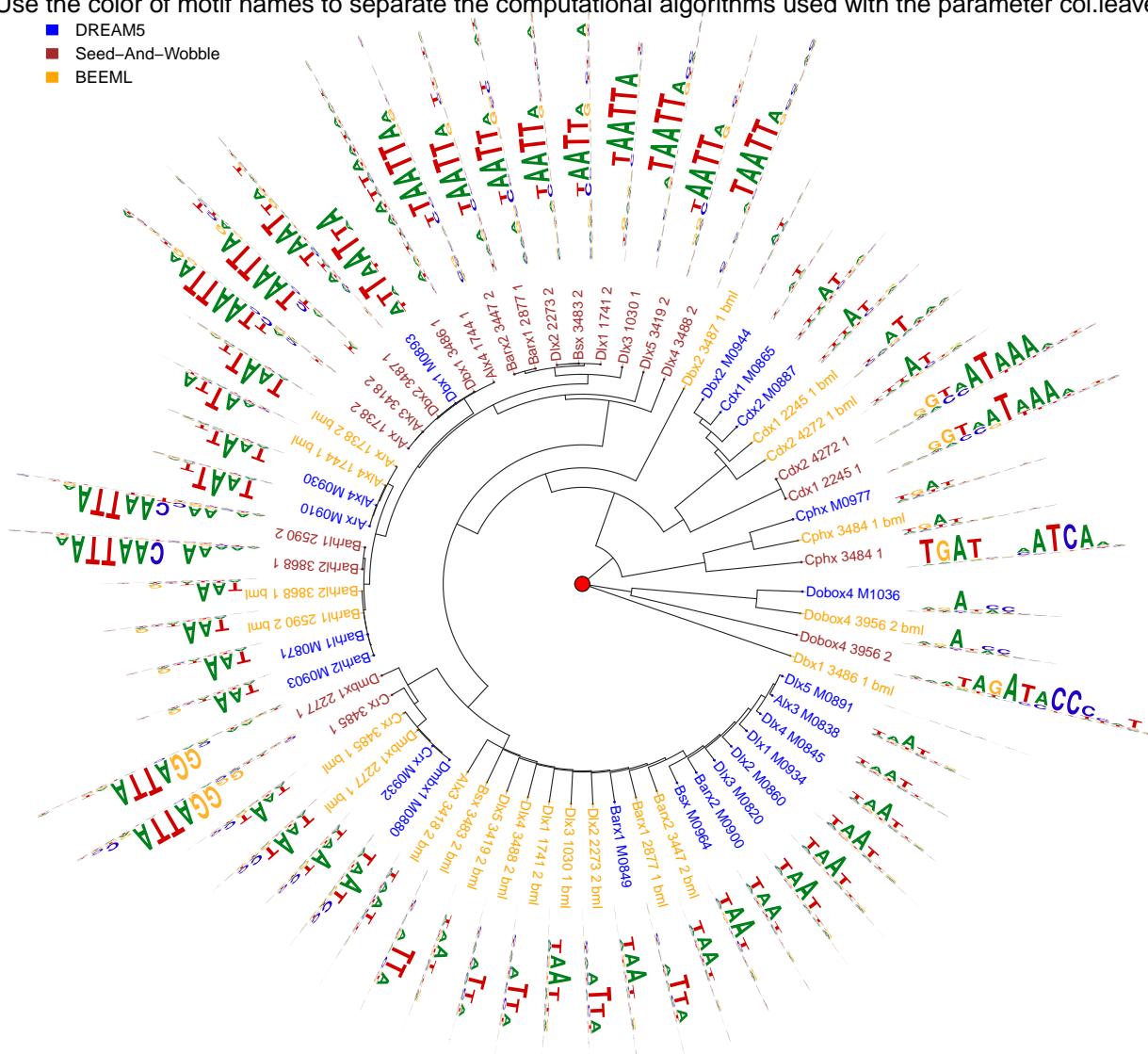
```
plotMotifStackWithRadialPhylog(phylog=phylog, pfms=unaligned.pfms,
                                labels.leaves=leaveNames,
                                col.leaves=algorithm,
                                cleaves=.2, circle=1.1, circle.motif=1.6,
                                clabel.leaves=.8, angle=358)
```

```
legend(-2.3, 2.4, legend=c("DREAM5", "Seed-And-Wobble", "BEEML"),
       fill= c("blue", "brown", "orange"),
       border="white", lty=NULL, bty = "n", cex=1)
```

```
text(0, 2.4, "Use the color of motif names to separate the computational algorithms used with the parameter col.leaves")
```

Use the color of motif names to separate the computational algorithms used with the parameter col.leaves

- DREAM5
- Seed-And-Wobble
- BEEML



Supplementary Figure 2I: color setting, leaves

Use the color of the background of the motif names to distinguish motifs with high and low IC with the parameter `col.leaves.bg`

```
icgp <- ifelse(sapply(sapply(unaligned.pfms, getIC), mean) > 0.6,
                "lightgray", "black")
plotMotifStackWithRadialPhylog(phylog=phylog, pfms=unaligned.pfms,
                                labels.leaves=leaveNames,
                                col.leaves.bg=icgp,
                                col.leaves.bg.alpha=.3,
                                cleaves=.2, circle=1.1, circle.motif=1.6,
                                clabel.leaves=.8, angle=358)
legend(-2.3, 2.4,
       legend=c("Information Content <= 0.6", "Information Content > 0.6"),
       fill= highlightCol(c("black", "lightgray"), alpha=.3),
       border="white", lty=NULL, bty = "n", cex=1)
text(0, 2.4,
     "Use the color of the background of the motif names to distinguish motifs with high and low IC with the parameter col.leaves.bg", cex=1.5)
```

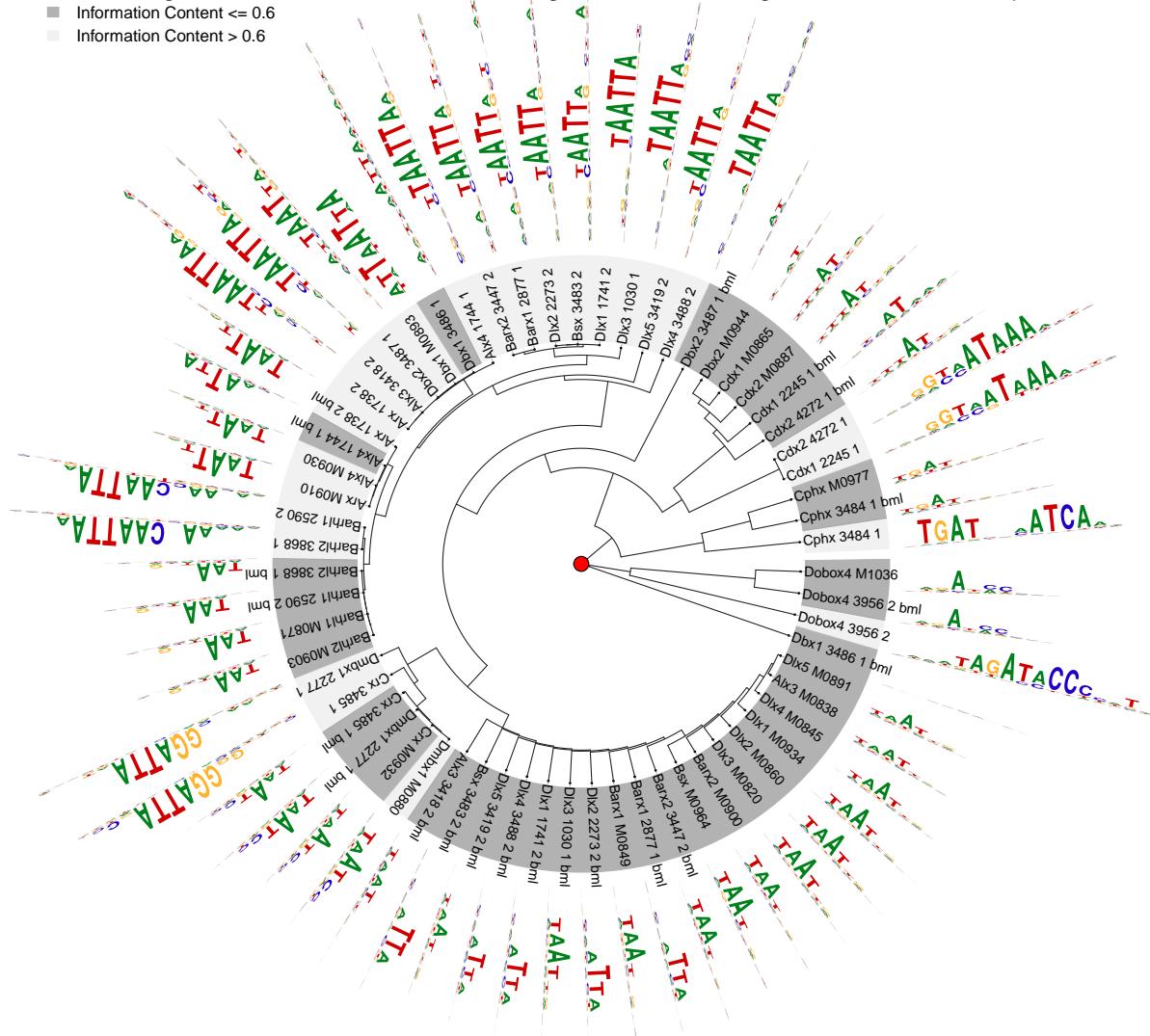
Use the color of the inner ring of the motif circle to distinguish motif groups with the parameter `col.inner.label.circle`

```
plotMotifStackWithRadialPhylog(phylog=phylog, pfms=unaligned.pfms,
                                labels.leaves=leaveNames,
                                col.inner.label.circle=motifGroup,
                                inner.label.circle.width=0.1,
                                cleaves=.2, circle=1.1, circle.motif=1.6,
                                clabel.leaves=.8, angle=358)
legend(1.5, 2.4, legend=levels.motifGroup[1:12],
       fill= colors.motifGroup[1:12],
       border="black", lty=NULL, bty = "n", ncol=2, cex=.8)
legend(1.5, -2, legend=levels.motifGroup[13:21],
       fill= colors.motifGroup[13:21],
       border="black", lty=NULL, bty = "n", ncol=2, cex=.8)
text(0, 2.4,
     "Use the color of the inner ring of the motif circle to distinguish motif groups with the parameter col.inner.label.circle", cex=1.5)
```

Use the color of the outer ring of the motif circle to distinguish motif groups with the parameter `col.outer.label.circle`

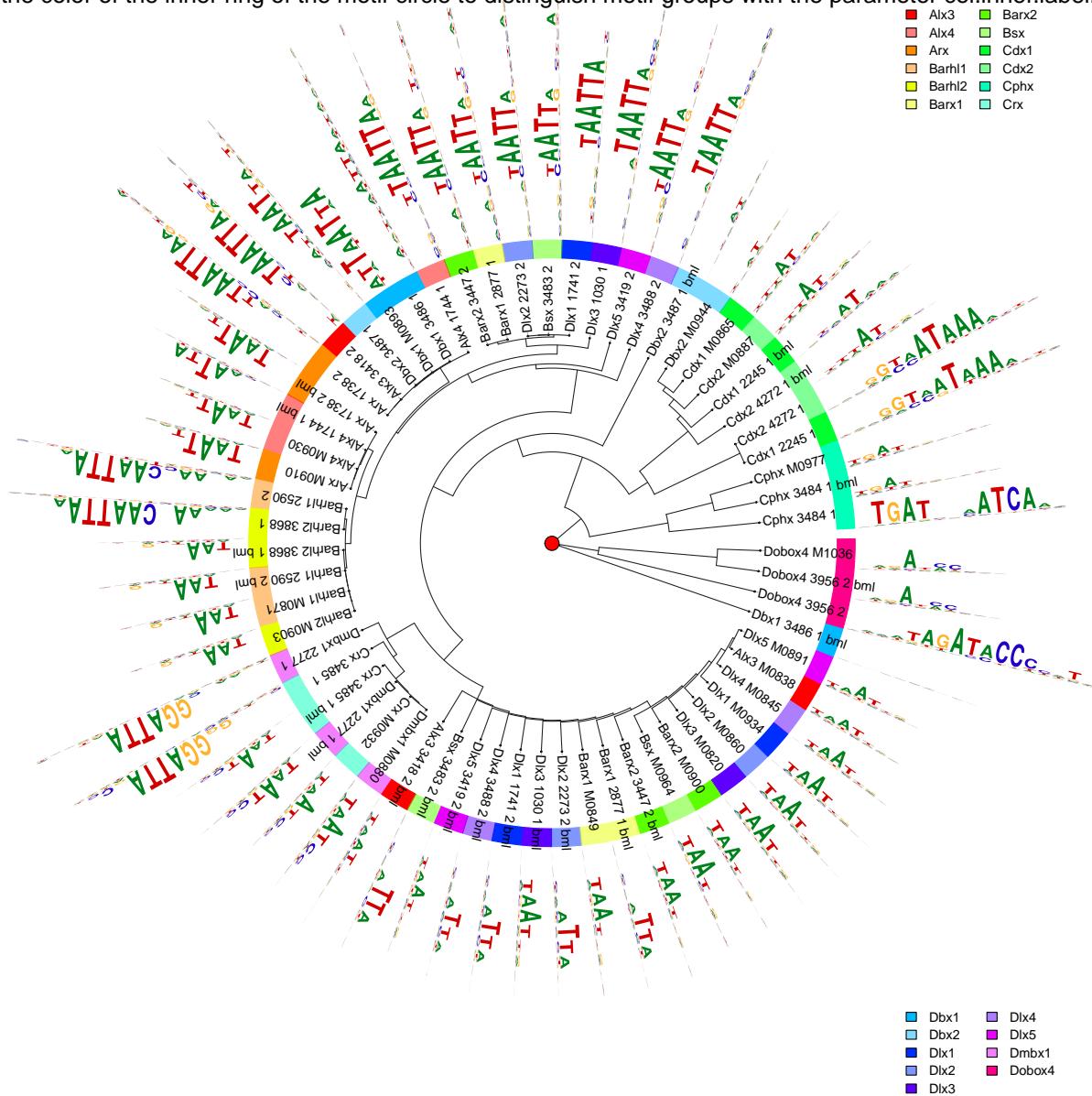
```
plotMotifStackWithRadialPhylog(phylog=phylog, pfms=unaligned.pfms,
                                labels.leaves=leaveNames,
                                col.outer.label.circle=motifGroup,
                                outer.label.circle.width=0.1,
                                cleaves=.2, circle=1.1, circle.motif=1.6,
                                clabel.leaves=.8, angle=358)
legend(1.5, 2.4, legend=levels.motifGroup[1:12],
       fill= colors.motifGroup[1:12],
       border="black", lty=NULL, bty = "n", ncol=2, cex=.8)
legend(1.5, -2, legend=levels.motifGroup[13:21],
```

color of the background of the motif names to distinguish motifs with high and low IC with the parameter col.le



Supplementary Figure 2J: color setting, leave background

the color of the inner ring of the motif circle to distinguish motif groups with the parameter col.inner.label.circle

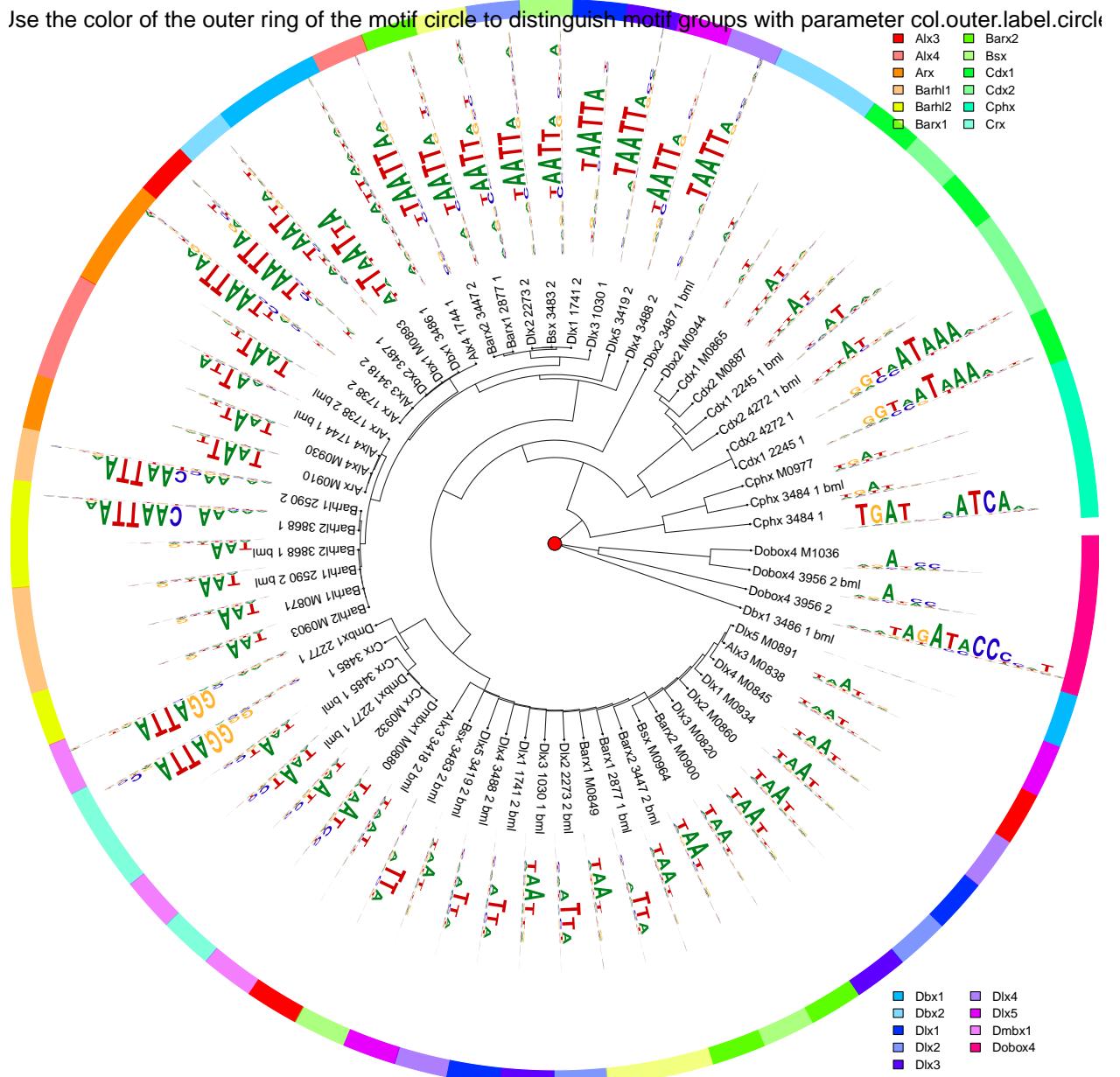


Supplementary Figure 2K: color setting, inner circle

```

fill= colors.motifGroup[13:21],
border="black", lty=NULL, bty = "n", ncol=2, cex=.8)
text(0, 2.4,
  "Use the color of the outer ring of the motif circle to distinguish motif groups with parameter col."
  "cex=1.5)

```



Supplementary Figure 2L: color setting, outer circle

Use a combination of coloring options to visualize different source of variations

```

plotMotifStackWithRadialPhylog(phylog=phylog, pfms=aligned.pfms,
                                labels.leaves=leaveNames,

```

```

    col.bg=algorithm, col.bg.alpha=.3,
    col.leaves.bg=icgp,
    col.leaves.bg.alpha=.3,
    cleaves=.2, circle=1.1, circle.motif=1.6,
    clabel.leaves=.8, angle=358)
legend(-2.3, 2.4,
       legend=c("algorithm", "CIS-BP/DREAM5", "Uniprobe/Seed-And-Wobble",
               "Uniprobe/BEEML", "Information Content", "<= 0.6", "> 0.6"),
       fill= c("white",
              highlightCol(c("blue", "brown", "orange"), alpha=.3),
              "white", highlightCol(c("black", "lightgray"), alpha=.3)),
       border="white", lty=NULL, bty = "n", cex=.8)

detach(motIVout)

```

Supplementary Figure 2N-O. Compare different Column Comparison Metrics (CCM) and alignment methods

Here we illustrate how to set the CCM and alignment method using MotIV or MatAlign, and that different choices of methods might affect the subsequent analysis and interpretation in motifStack.

When clustering using MotIV, Pearson Correlation Coefficient (PCC) or Average Log Likelihood Ratio (ALLR) is used as the CCM, and Smith-Waterman Ungapped (SWU) or Needleman-Wunsch (NW) is used as the alignment method. When clustering the motifs using MatAlign, ALLR and SWU are used.

To run the MatAlign examples, phylib (<http://evolution.genetics.washington.edu/phylip/progs.data.dist.html>) and MatAlign (<http://stormo.wustl.edu/MatAlign/>) need to be installed first.

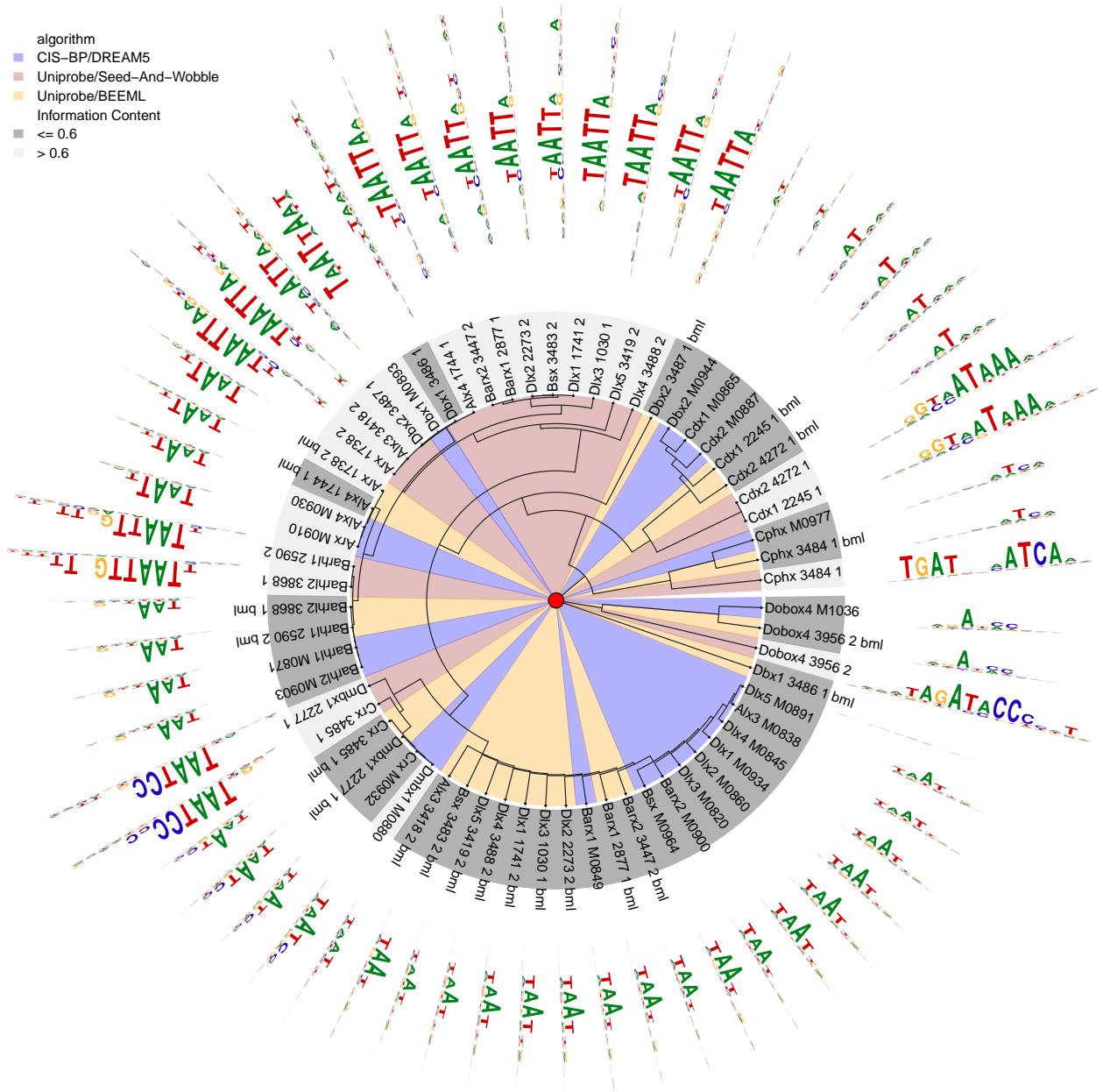
Visualize motif alignments using MotIV with different CCM and alignment methods

```

sta <- mapply(function(cc, align){
  motIVout <- getMotIVOut(pfms, cc, align)
  attach(motIVout)
  plotMotifStackWithRadialPhylog(phylog=phylog, pfms=aligned.pfms,
                                  labels.leaves=leaveNames,
                                  col.bg=algorithm, col.bg.alpha=.3,
                                  col.inner.label.circle=motifGroup,
                                  inner.label.circle.width=0.1,
                                  cleaves=.2, circle=1.1, circle.motif=1.6,
                                  clabel.leaves=.3, angle=358)

  legend(-2.3, 2.4,
         legend=c("CIS-BP/DREAM5", "Uniprobe/Seed-And-Wobble",
                 "Uniprobe/BEEML"),
         fill= highlightCol(c("blue", "brown", "orange"), alpha=.3),
         border="white", lty=NULL, bty = "n", cex=.5)
  text(0, 2.3, label=paste("cc=", cc, "; align=", align), cex=1.5)
  cnt <- rle(motifGroup)
  cnt <- split(algorithm, rep(1:length(cnt$lengths), cnt$lengths))
  cnt <- table(sapply(cnt, function(.ele) length(unique(.ele))))
  cnt.1 <- vector("integer", 3)
  names(cnt.1) <- 1:3
}

```

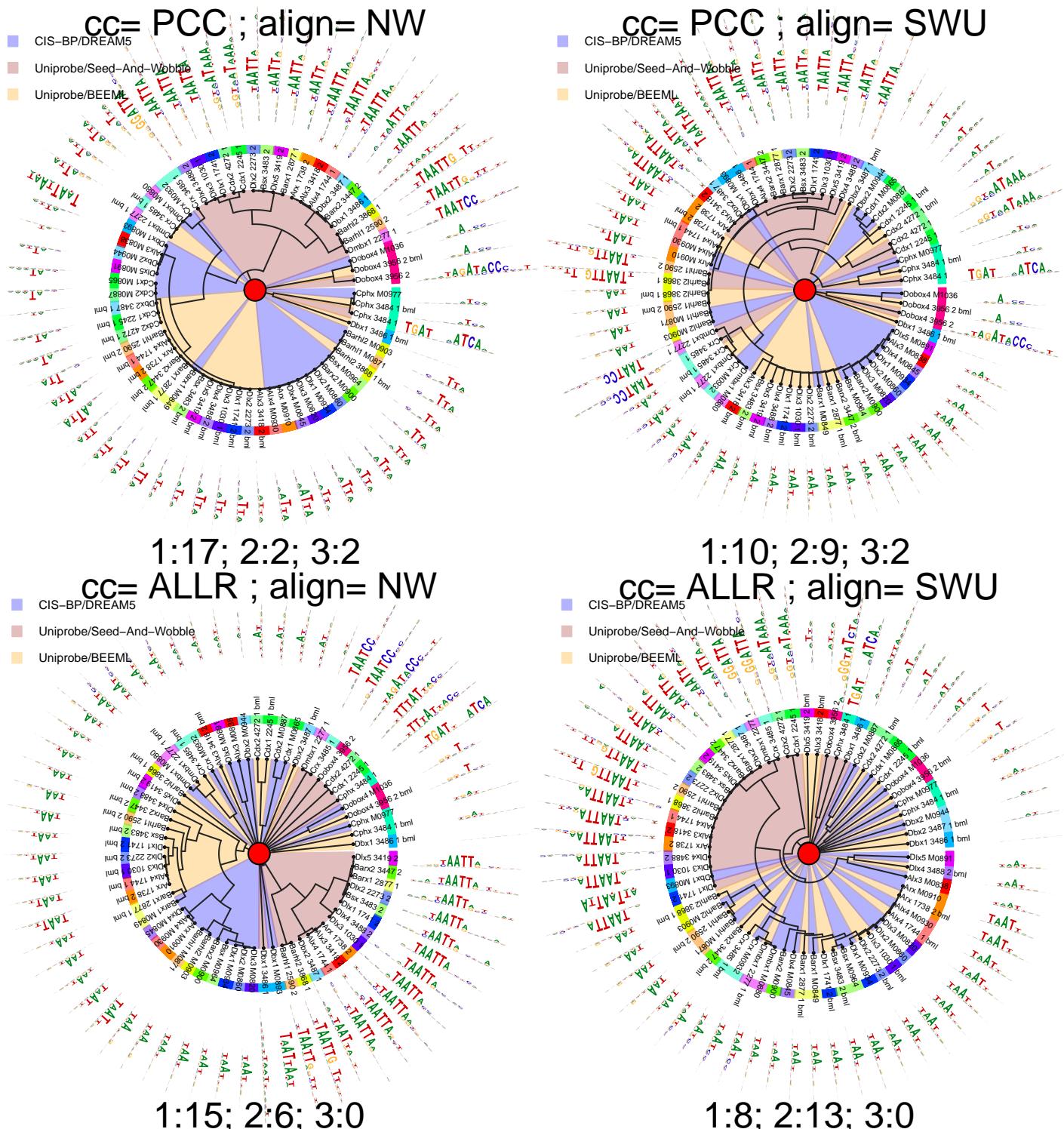


Supplementary Figure 2M: color setting, use a combination of coloring options

```

cnt.1[names(cnt)] <- cnt
cnt.1["1"] <- (cnt.1["1"]+cnt.1["2"]*2+cnt.1["3"]*3)/3 - cnt.1["2"] - cnt.1["3"]
text(0, -2.3,
     label=paste(names(cnt.1), cnt.1, sep=":", collapse="; "), cex=1.5)
detach(motIVout)
}, c("PCC", "PCC", "ALLR", "ALLR"), c("NW", "SWU", "NW", "SWU"))

```



The number (1:xx; 2:yy; 3:zz) on the bottom of the figure 2O and 2P represents the number of motifs identified by different number of motif comparison and clustering algorithms. 1:xx means that xx number of motifs are identified by one algorithm; 2:yy means that yy number of motifs are identified by two different algorithms; 3:zz means that zz number of motifs are identified by three different algorithms.

Visualize motif alignments using MatAlign with ALLR as the MCC and SWU as the alignment method

```

##read newick tree. Alignment is done by MatAlign
##and the newick tree is generated by Neighbor, which is a part of phylip
source("zzz.R")
system(paste("perl", MatAlign2tree_path, "--in . --pcmpath", pcmpath,
            "--out", outpath,
            "--matalign", matalign_path,
            "--neighbor", neighbor_path,
            "--tree","UPGMA"))
newickstrUPGMA <- readLines(con=file.path(outpath, "NJ.matalign.distMX.nwk"))
##convert it to phylog object
phylogUPGMAmatAlign <- newick2phylog(newickstrUPGMA, FALSE)

##get the leaves of phylog for reordering the pfms
leaveNames <- names(phylogUPGMAmatAlign$leaves)
this_motifs <- pfms[leaveNames]

## data source
dataSource <- factor(grep("_M", leaveNames))
levels(dataSource) <- c("yellow", "blue") ##c("Uniprobe", "CIS-BP")
dataSource <- as.character(dataSource)
## algorithm
algorithm <- factor(!grep("_M", leaveNames)) + grep("_bml", leaveNames)
levels(algorithm) <- c("blue", "brown", "orange") ##("DREAM5", "Seed-And-Wobble", "BEEML")
algorithm <- as.character(algorithm)
## motifs from the same PBM data source
motifGroup <- factor(gsub("(.*?)_.*$", "\\\1", leaveNames))
levels.motifGroup <- levels(motifGroup)
levels(motifGroup) <- pairColor[1:length(levels(motifGroup))]
colors.motifGroup <- levels(motifGroup)
motifGroup <- as.character(motifGroup)

## draw the motifs
plotMotifStackWithRadialPhylog(phylog=phylogUPGMAmatAlign,
                                 pfms=DNAmotifAlignment(this_motifs),
                                 labels.leaves=leaveNames,
                                 col.bg=algorithm,
                                 col.bg.alpha=.3,
                                 col.inner.label.circle=motifGroup,
                                 inner.label.circle.width=.1,
                                 cleaves=.2, circle=1.1,
                                 circle.motif=1.6,
                                 clabel.leaves=.6, angle=358)

legend(1.5, 2.4, legend=levels.motifGroup[1:12],
       fill= colors.motifGroup[1:12],

```

```

border="black", lty=NULL, bty = "n", ncol=2, cex=.8)
legend(1.5, -2, legend=levels.motifGroup[13:21],
       fill= colors.motifGroup[13:21],
       border="black", lty=NULL, bty = "n", ncol=2, cex=.8)
legend(-2.35, 2.4,
       legend=c("CIS-BP/DREAM5", "Uniprobe/Seed-And-Wobble", "Uniprobe/BEEML"),
       fill= highlightCol(c("blue", "brown", "orange"), alpha=.3),
       border="white", lty=NULL, bty = "n", cex=.8)
cnt <- rle(motifGroup)
cnt <- split(algorithm, rep(1:length(cnt$lengths), cnt$lengths))
cnt <- table(sapply(cnt, function(.ele) length(unique(.ele))))
cnt.1 <- vector("integer", 3)
names(cnt.1) <- 1:3
cnt.1[names(cnt)] <- cnt
cnt.1["1"] <- (cnt.1["1"]+cnt.1["2"]*2+cnt.1["3"]*3)/3 - cnt.1["2"] - cnt.1["3"]
text(0, -2.3, label=paste(names(cnt.1), cnt.1, sep=":", collapse="; "), cex=1.5)

```

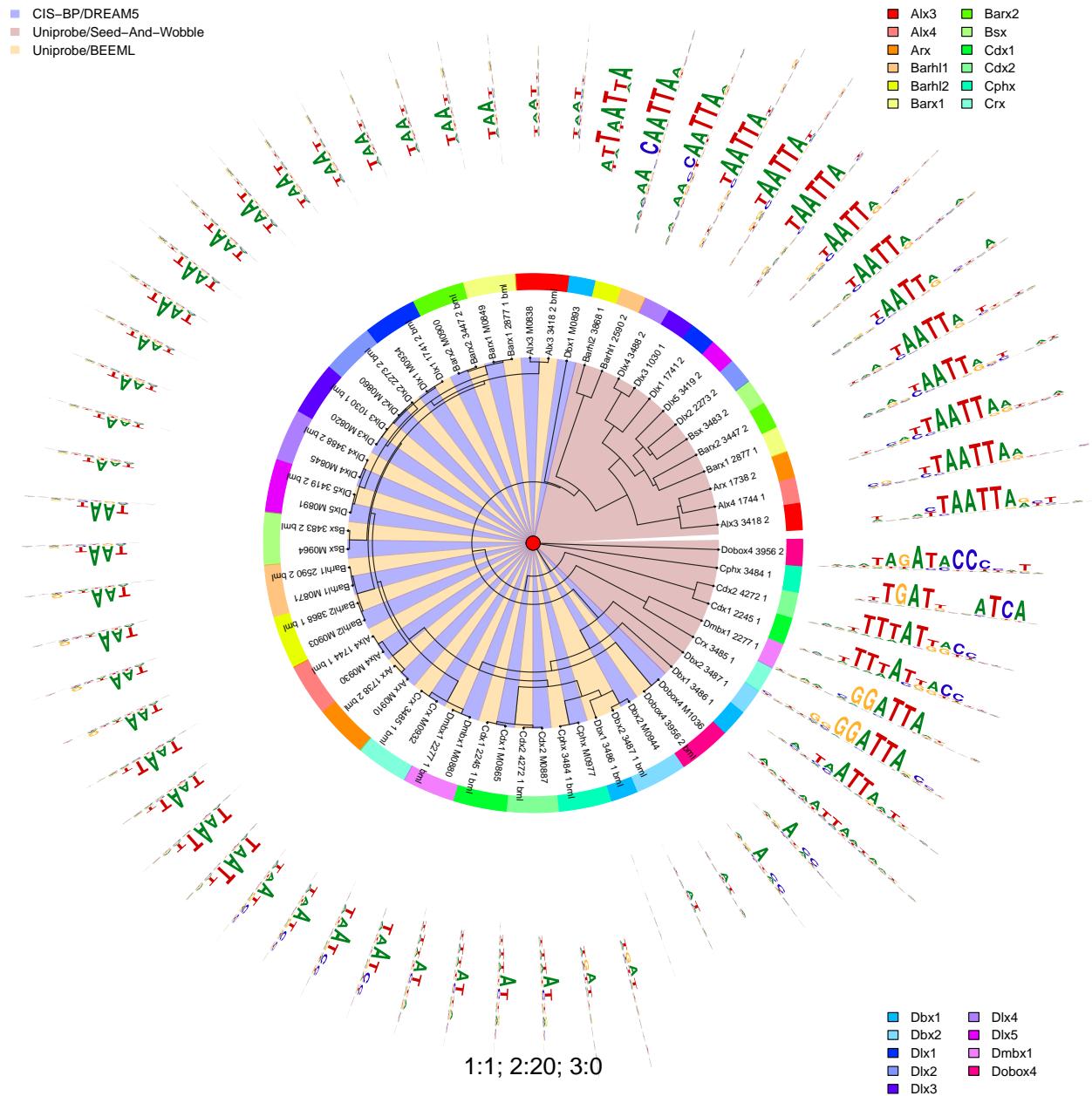
Supplementary Figure 2P. Different distance resolution between MotIV and MatAlign even when using the same Column Comparison Matrix and alignment method

```

pcmpath <- "pcmsDatasetDM"
pcms <- readPCM(pcmpath)
pfms<-lapply(pcms,pcm2pfm)
motIVout <- getMotIVOut(pfms, "ALLR", "SWU")
plotMotifStackWithRadialPhylog(phylog=motIVout$phylog,
                                pfms=motIVout$aligned.pfms,
                                labels.leaves=motIVout$leaveNames,
                                cleaves=.2, circle=1.2, circle.motif=1.6,
                                clabel.leaves=1,
                                motifScale="logarithmic",
                                angle=358,
                                plotIndex=FALSE)
text(0, 2.4, label="motIV: cc=ALLR; align=SWU", cex=1.5)

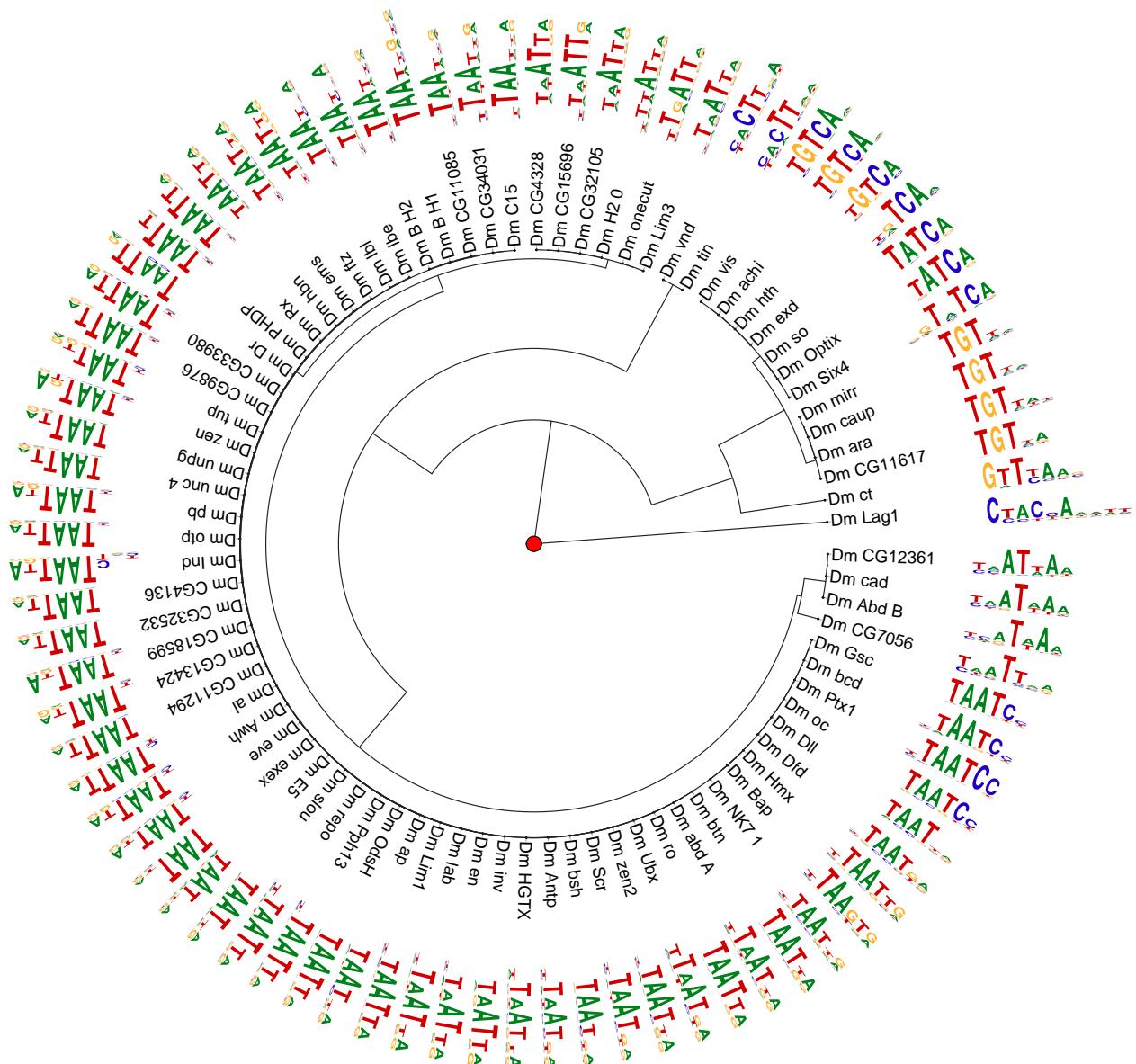
## function to read example data
getMatAlignOut <- function(pcmpath, groupDistance=NA, trim=0.2){
  pcms <- readPCM(pcmpath)
  pfms<-lapply(pcms,pcm2pfm)
  source("zzz.R")
  system(paste("perl", MatAlign2tree_path, "--in . --pcmpath", pcmpath,
              "--out", outpath,
              "--matalign", matalign_path,
              "--neighbor", neighbor_path,
              "--tree", "UPGMA"))
  newickstrUPGMA <-
    readLines(con=file.path(outpath, "NJ.matalign.distMX.nwk"))
  phylog <- newick2phylog(newickstrUPGMA, FALSE)
  leaves <- names(phylog$leaves)
  motifs <- pfms[leaves]
  if(!is.na(groupDistance)){
    motifSig <-

```



Supplementary Figure 2O: UPGMA tree by matAlign+PHYLIP

motIV: cc=ALLR; align=SWU



Supplementary Figure 2P.a: MotIV(ALLR and SWU) as the alignment method

```

        motifSignature(motifs, phylog,
                        groupDistance=groupDistance,
                        min.freq=1, trim=trim)
    sig <- signatures(motifSig)
    gpCol <- sigColor(motifSig)
} else{
    motifSig <- NA
    sig <- NA
    gpCol <- NA
}

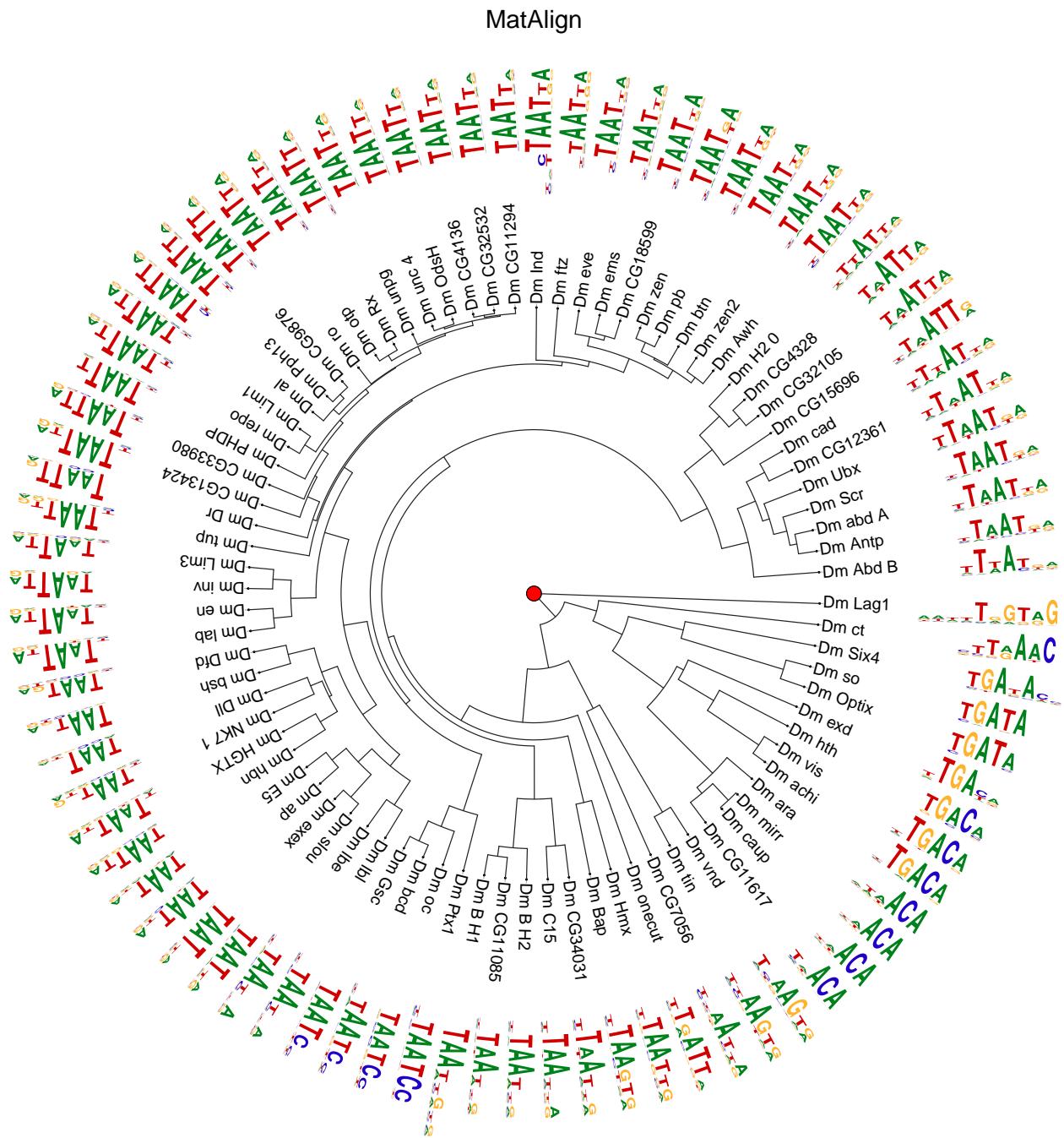
return(list(phylog=phylog, sig=sig, gpCol=gpCol,
            motifs=DNAmotifAlignment(motifs),
            leaves=leaves,
            unaligned.pfms=motifs))
}

matAlignOut <- getMatAlignOut(pcopath)
plotMotifStackWithRadialPhylog(phylog=matAlignOut$phylog,
                                pfms=matAlignOut$motifs,
                                labels.leaves=matAlignOut$leaves,
                                cleaves=.2, circle=1.2, circle.motif=1.6,
                                clabel.leaves=1, motifScale="logarithmic",
                                angle=358, plotIndex=FALSE)
text(0, 2.4, label="MatAlign", cex=1.5)

sessionInfo()

## R version 3.4.1 (2017-06-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: OS X El Capitan 10.11.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4     parallel   grid       stats      graphics   grDevices utils
## [8] datasets   methods    base
##
## other attached packages:
## [1] motifStack_1.20.1   Biostrings_2.44.2   XVector_0.16.0
## [4] IRanges_2.10.2     S4Vectors_0.14.3   ade4_1.7-6
## [7] MotIV_1.32.0       BiocGenerics_0.22.0 grImport_0.9-0
## [10] XML_3.98-1.9
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.12          highr_0.6
## [3] plyr_1.8.4            compiler_3.4.1
## [5] GenomeInfoDb_1.12.2   bitops_1.0-6
## [7] tools_3.4.1           zlibbioc_1.22.0

```



Supplementary Figure 2P.b: MatAlign (ALLR and SWU) as the alignment method

```
## [9] digest_0.6.12           evaluate_0.10.1
## [11] lattice_0.20-35         BSgenome_1.44.0
## [13] Matrix_1.2-10          DelayedArray_0.2.7
## [15] yaml_2.1.14             seqLogo_1.42.0
## [17] GenomeInfoDbData_0.99.0 rtracklayer_1.36.4
## [19] stringr_1.2.0           knitr_1.16
## [21] htmlwidgets_0.9          rprojroot_1.2
## [23] Biobase_2.36.2          BiocParallel_1.10.1
## [25] rGADEM_2.24.0           rmarkdown_1.6
## [27] magrittr_1.5              scales_0.4.1
## [29] backports_1.1.0          Rsamtools_1.28.0
## [31] htmltools_0.3.6           matrixStats_0.52.2
## [33] GenomicRanges_1.28.4      GenomicAlignments_1.12.1
## [35] SummarizedExperiment_1.6.3 colorspace_1.3-2
## [37] stringi_1.1.5            munsell_0.4.3
## [39] RCurl_1.95-4.8
```