Figures for Paper

Awais

15 February 2018

```
library(readr)
library(VennDiagram)
```

```
## Loading required package: grid
```

```
## Loading required package: futile.logger
```

```
library(magrittr)
library(rtracklayer)
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colMeans,
##     colnames, colSums, do.call, duplicated, eval, evalq, Filter,
##     Find, get, grep, grepl, intersect, is.unsorted, lapply,
##     lengths, Map, mapply, match, mget, order, paste, pmax,
##     pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,
##     rowMeans, rownames, rowSums, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:base':
##
##     expand.grid
```

```
## Loading required package: IRanges
```

```
## Loading required package: GenomeInfoDb
```

```
library(pander)
```

```
ChIPSeqEnrichedRegions <- read_delim("/media/awais/NewDrivewho/Downloads/IDR_final_conservative.narrowP
                                     "\t", escape_double = FALSE,col_names = FALSE, trim_ws = TRUE)


MotifOverlapRPredictedSites <- read_delim("/media/awais/NewDrivewho/ChangesToMotifOverlapR/Gata4HEP2Full
                                  "\t", escape_double = FALSE, col_names = FALSE,
                                  trim_ws = TRUE)


# CRM motifs

CRMMotifInstances <- read_delim("/media/awais/NewDrivewho/ChangesToMotifOverlapR/GATA4CRMMotifs.txt",
                                "\t", escape_double = FALSE, col_names = FALSE,
                                trim_ws = TRUE)


AllMotifInstances <- read_delim("/media/awais/NewDrivewho/ChangesToMotifOverlapR/GATA4HEP2AllMotifs",
                                "\t", escape_double = FALSE, col_names = FALSE,
                                trim_ws = TRUE)


enhancers <- import("/media/awais/NewDrivewho/Downloads/human_permissive_enhancers_phase_1_and_2.bed.gz
promoters <- import("/media/awais/NewDrivewho/Downloads/hg.bed.gz")%>%promoters()
```

Converting the Dataframes to Genomic Ranges

```
ChIPSeqEnrichedRegionsGRanges <- makeGRangesFromDataFrame(ChIPSeqEnrichedRegions,
                         keep.extra.columns=TRUE,
                         ignore.strand=TRUE,
                         seqinfo=NULL,
                         seqnames.field="X1",
                         start.field="X2",
                         end.field="X3",
                         strand.field="X8",
                         starts.in.df.are.0based=FALSE)

# Expaning the ChIP-enriched region by 200bp up and down stream to identify motifs in surrounding areas

ChIPSeqEnrichedRegionsGRanges <- ChIPSeqEnrichedRegionsGRanges+200

MotifOverlapRPredictedSitesGRanges <- makeGRangesFromDataFrame(MotifOverlapRPredictedSites,
                         keep.extra.columns=TRUE,
                         ignore.strand=TRUE,
                         seqinfo=NULL,
                         seqnames.field="X1",
                         start.field="X2",
                         end.field="X3",
                         strand.field="X5",
                         starts.in.df.are.0based=FALSE)%>%reduce()
```

```
AllMotifInstancesGRanges <- makeGRangesFromDataFrame(AllMotifInstances,
                                                     keep.extra.columns=TRUE,
                                                     ignore.strand=TRUE,
                                                     seqinfo=NULL,
                                                     seqnames.field="X1",
                                                     start.field="X2",
                                                     end.field="X3",
                                                     strand.field="X5",
                                                     starts.in.df.are.0based=FALSE)


CRMMotifInstancesGRanges <- makeGRangesFromDataFrame(CRMMotifInstances,
                                                     keep.extra.columns=TRUE,
                                                     ignore.strand=TRUE,
                                                     seqinfo=NULL,
                                                     seqnames.field="X1",
                                                     start.field="X2",
                                                     end.field="X3",
                                                     strand.field="X5",
                                                     starts.in.df.are.0based=FALSE)
```

Lets get the overlaps between Predicted sites and ChIP-enriched regions within Regulatory modules

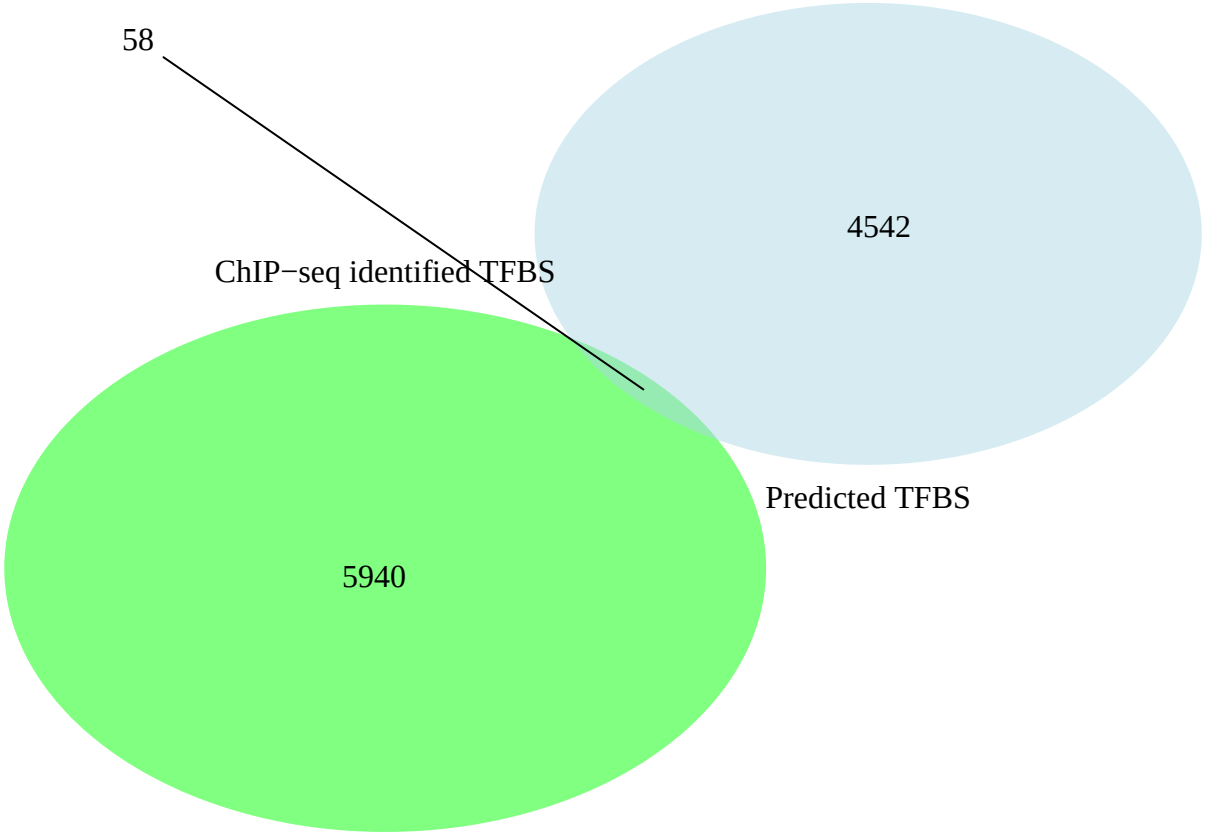
We select for ChIP-enriched regions within Regulatory modules as these regions are usually the ones that affect gene regulation and therefore the ones we're interested/

```
## Identifing ChIP-seq regions in regulatory modules
#
# SitesInRegulatoryModules <- c(subsetByOverlaps(ChIPSeqEnrichedRegionsGRanges, enhancers),
#                               subsetByOverlaps(ChIPSeqEnrichedRegionsGRanges, promoters) )%>% unlist(


SitesInRegulatoryModules <- ChIPSeqEnrichedRegionsGRanges
OverlapPredictedChIPseq <- (findOverlaps(MotifOverlapRPredictedSitesGRanges, SitesInRegulatoryModules)%

OverlapSequneceChIPseq <- (findOverlaps(AllMotifInstancesGRanges, SitesInRegulatoryModules)%>%countLnode

OverlapPredictedSequence <- (findOverlaps(MotifOverlapRPredictedSitesGRanges, AllMotifInstancesGRanges)

OverlapCRMEnriched <- (findOverlaps(CRMMotifInstancesGRanges, SitesInRegulatoryModules)%>%countLnodeHit


OverlapALL <- (subsetByOverlaps(subsetByOverlaps(MotifOverlapRPredictedSitesGRanges, AllMotifInstancesG
```

Lets make some pretty diagrams

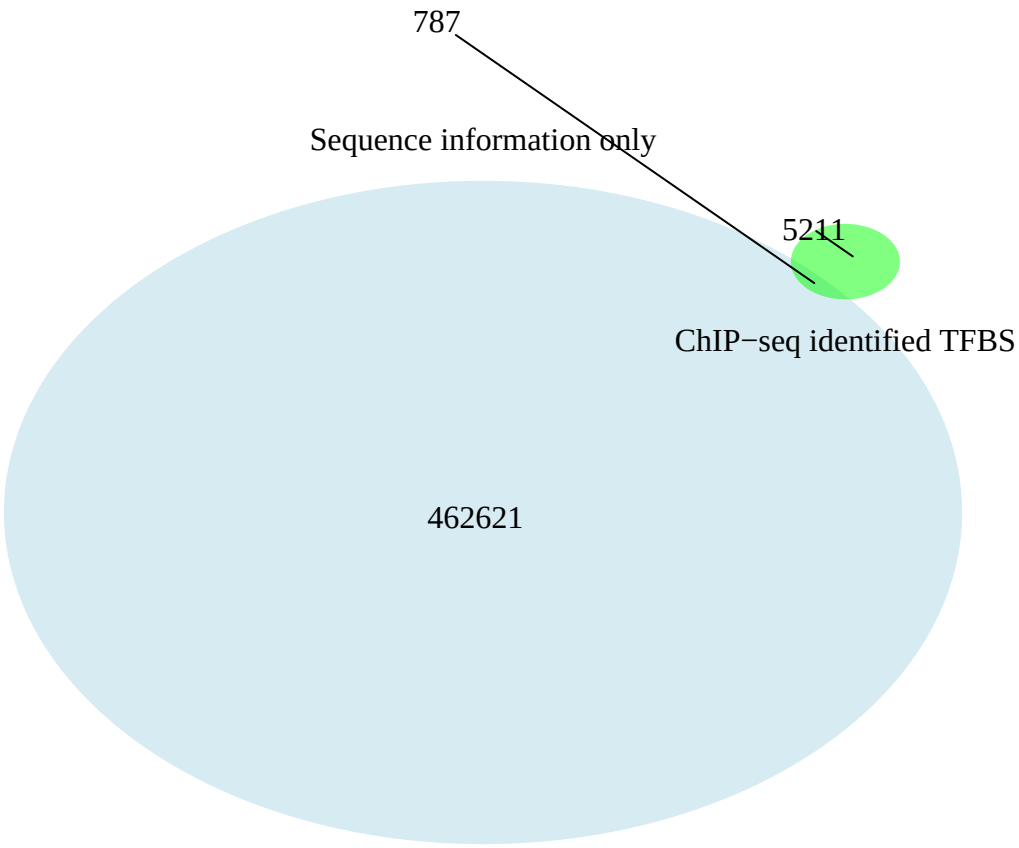
```
grid.newpage()
draw.pairwise.venn(length(MotifOverlapRPredictedSitesGRanges),
```

```
                  length(SitesInRegulatoryModules),
                  OverlapPredictedChIPseq,
                  category = c("Predicted TFBS", "ChIP-seq identified TFBS"),
                  lty = rep("blank", 2),
                  fill = c("light blue", "green"),
                  alpha = rep(0.5, 2),
                  cat.pos = c(0, 180),
                  euler.d = TRUE,
                  sep.dist = 0.03,
                  rotation.degree = 45)
```



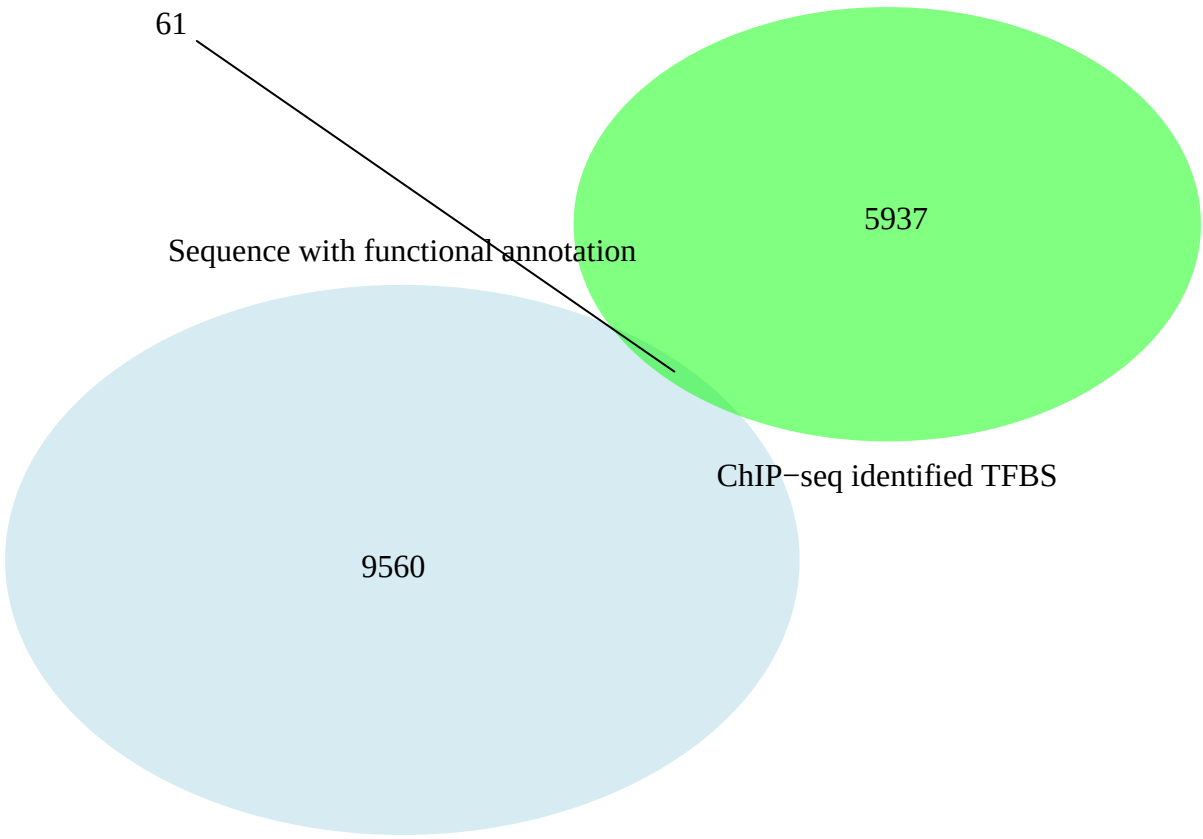

```
## (polygon[GRID.polygon.11], polygon[GRID.polygon.12], polygon[GRID.polygon.13], polygon[GRID.polygon.
```

```
grid.newpage()
draw.pairwise.venn(length(AllMotifInstancesGRanges),
                  length(SitesInRegulatoryModules),
                  OverlapSequneceChIPseq,
                  category = c("Sequence information only", "ChIP-seq identified TFBS"),
                  lty = rep("blank", 2),
                  fill = c("light blue", "green"),
                  alpha = rep(0.5, 2),
                  cat.pos = c(0, 180),
                  euler.d = TRUE,
                  sep.dist = 0.03,
                  rotation.degree = 45)
```




```
## (polygon[GRID.polygon.21], polygon[GRID.polygon.22], polygon[GRID.polygon.23], polygon[GRID.polygon.1
```

```
grid.newpage()
draw.pairwise.venn(length(CRMMotifInstancesGRanges),
                   length(SitesInRegulatoryModules),
                   OverlapCRMEnriched,
                   category = c("Sequence with functional annotation", "ChIP-seq identified TFBS"),
                   lty = rep("blank", 2),
                   fill = c("light blue", "green"),
                   alpha = rep(0.5, 2),
                   cat.pos = c(0, 180),
                   euler.d = TRUE,
                   sep.dist = 0.03,
                   rotation.degree = 45)
```




```
## (polygon[GRID.polygon.32], polygon[GRID.polygon.33], polygon[GRID.polygon.34], polygon[GRID.polygon.3
```

```
ImprovedAccuracyRelativeToSequence <-
  function(x){

    foldImprovement <- (subsetByOverlaps(x, SitesInRegulatoryModules)%>%length() / x%>%length()) /
    (subsetByOverlaps(AllMotifInstancesGRanges, SitesInRegulatoryModules)%>%length () / AllMotifInstanc


  }

PercentageOfSitesThatAreTruePositives<- function(x) {
   (subsetByOverlaps(x, SitesInRegulatoryModules)%>%length() / x%>%length())*100
}

AccuracyRelativeToSequence <- lapply(list(AllMotifInstancesGRanges,
         CRMMotifInstancesGRanges,
         MotifOverlapRPredictedSitesGRanges),
       ImprovedAccuracyRelativeToSequence)%>%rbind.data.frame()%>%set_colnames(c("Sequence", "Functiona

cbind.data.frame("Sequence" = c(
                 length(AllMotifInstancesGRanges),
                 length(ChIPSeqEnrichedRegionsGRanges),
                 OverlapSequneceChIPseq,
                 PercentageOfSitesThatAreTruePositives(AllMotifInstancesGRanges),
                 ImprovedAccuracyRelativeToSequence(AllMotifInstancesGRanges)),
                 "CRM Sites" =
                   c(
```

```
                length(CRMMotifInstancesGRanges),
                length(ChIPSeqEnrichedRegionsGRanges),
                OverlapCRMEnriched,
                PercentageOfSitesThatAreTruePositives(CRMMotifInstancesGRanges),
                ImprovedAccuracyRelativeToSequence(CRMMotifInstancesGRanges)),
                "motifOverlapR Sites" = c(
                length(MotifOverlapRPredictedSitesGRanges),
                length(ChIPSeqEnrichedRegionsGRanges),
                OverlapPredictedChIPseq,
                PercentageOfSitesThatAreTruePositives(MotifOverlapRPredictedSitesGRanges),
                ImprovedAccuracyRelativeToSequence(MotifOverlapRPredictedSitesGRanges))
                ) %>%set_rownames(c ("Number of motifs",
                                     "Number of ChIP-seq sites",
                                     "Number of sites Predicted",
                                     "% of predicted sites that are true positives",
                                     "Fold enrichement over sequence alone")
                  )%>%pander()
```

	Sequence	CRM Sites	motifOverlapR Sites
Number of motifs	463408	9621	4600
Number of ChIP-seq sites	5998	5998	5998
Number of sites Predicted	787	61	58
% of predicted sites that are true positives	0.1698	0.634	1.261
Fold enrichement over sequence alone	1	3.733	7.424