PepBed examples (Human)

November 19, 2017

Parsing bigbed files

```
library(PepBed)

# path to bigbed file(s)
bb_path <- '/home/biolinux/temp/human/pride_cluster_peptides_9606_Human_pogo.bb'
bb_mod_path <- '/home/biolinux/temp/human/pride_cluster_peptides_9606_Human_pogo_ptm.bb'

# convert bigbed to bed file (output bed file in the same directory)
bigbed2bed(inputFile = bb_path, compress = FALSE)
bigbed2bed(inputFile = bb_mod_path, compress = FALSE)

# getting basic information (output description file in the same directory)
getBigBedInfo(inputFile = bb_path)
getBigBedInfo(inputFile = bb_mod_path)

# getting field names if available
fieldNames <- getBigBedFieldNames(inputFile = bb_path, only.names = TRUE)

print(fieldNames)
```

```
##  [1] "chrom"       "chromStart"  "chromEnd"    "name"        "score"
##  [6] "strand"      "thickStart"  "thickEnd"    "reserved"    "blockCount"
## [11] "blockSizes"  "chromStarts"
```

Parsing Bed file

```
# path to bed file(s)
bed_path <- '/home/biolinux/temp/human/pride_cluster_peptides_9606_Human_pogo.bed'
bed_mod_path <-  '/home/biolinux/temp/human/pride_cluster_peptides_9606_Human_pogo_ptm.bed'

# import bed file as  dataframe
bed_df <- readBedFile(inputFile = bed_path)
bed_mod_df <- readBedFile(inputFile = bed_mod_path)

# set column name to bed file
names(bed_df) <- fieldNames
names(bed_mod_df) <- fieldNames

# convert dataframe to GRanges
# all non-modified peptides
granges_peptide <- buildGRangesFromData(data = bed_df,
                                        chrColName = "chrom",
                                        startColName = "chromStart",
                                        endColName = "chromEnd")

# all modified peptides
granges_mod_peptide <- buildGRangesFromData(data = bed_mod_df,
                                            chrColName = "chrom",
                                            startColName = "chromStart",
                                            endColName = "chromEnd")
```

Computing some basic stats from the data

```
# getting number of features(peptides) by chromosome
counts <- countsByChromosome(gr = granges_peptide, colName = 'Peptides')
counts_mod <- countsByChromosome(gr = granges_mod_peptide, colName = 'Peptides_mod')

# merging dfs
merged_counts <- merge.data.frame(counts, counts_mod, by = 'Chromosome')
# ordering by chromosome
merged_counts <- orderByChromosome(df = merged_counts, colName = 'Chromosome')

print(merged_counts)
```

```
##    Chromosome Peptides Peptides_mod
## 1           1     9736         6140
## 12          2     7490         4541
## 16          3     5200         2719
## 17          4     3505         2066
## 18          5     4013         2180
## 19          6     6463         6147
## 20          7     4121         2524
## 21          8     3064         1472
## 22          9     3841         2453
## 2          10     3688         2201
## 3          11     5991         3806
## 4          12     6191         4525
## 5          13     1183          710
## 6          14     3822         2223
## 7          15     2704         1784
## 8          16     3524         1977
## 9          17     6678         4220
## 10         18     1176          679
## 11         19     5536         3404
## 13         20     2264         1238
## 14         21      867          552
## 15         22     2236         1732
## 24          X     3138         2059
## 25          Y      131           72
## 23          M       19           14
```

Getting stats for unique peptides

```
# removing duplicated entries from original granges_peptide
unique_pep <- getUniqueFeatures(granges_peptide, colFeatures = 'name')
unique_pep_mod <- getUniqueFeatures(granges_mod_peptide, colFeatures = 'name')

# getting unique number of features(peptides) by chromosome
counts_unique <- countsByChromosome(gr = unique_pep, colName = 'Peptides')
counts_mod_unique <- countsByChromosome(gr = unique_pep_mod, colName = 'Peptides_mod')

# merging dfs
merged_counts_unique <- merge.data.frame(counts_unique,
                                         counts_mod_unique,
                                         by = 'Chromosome')

# ordering by chromosome
merged_counts_unique <- orderByChromosome(df = merged_counts_unique,
                                          colName = 'Chromosome')

print(merged_counts_unique)
```

```
##    Chromosome Peptides Peptides_mod
## 1           1     8801         4532
## 12          2     6073         2802
## 16          3     4622         2095
## 17          4     3145         1642
## 18          5     3385         1632
## 19          6     3759         1824
## 20          7     3338         1608
## 21          8     2543         1039
## 22          9     3224         1698
## 2          10     3379         1682
## 3          11     5341         2882
## 4          12     4985         2832
## 5          13     1031          460
## 6          14     3123         1609
## 7          15     2336         1237
## 8          16     2942         1388
## 9          17     5491         2761
## 10         18     1016          442
## 11         19     4666         2443
## 13         20     1935          881
## 14         21      666          313
## 15         22     1742         1084
## 24          X     2388         1310
## 25          Y        8            4
## 23          M       19           12
```

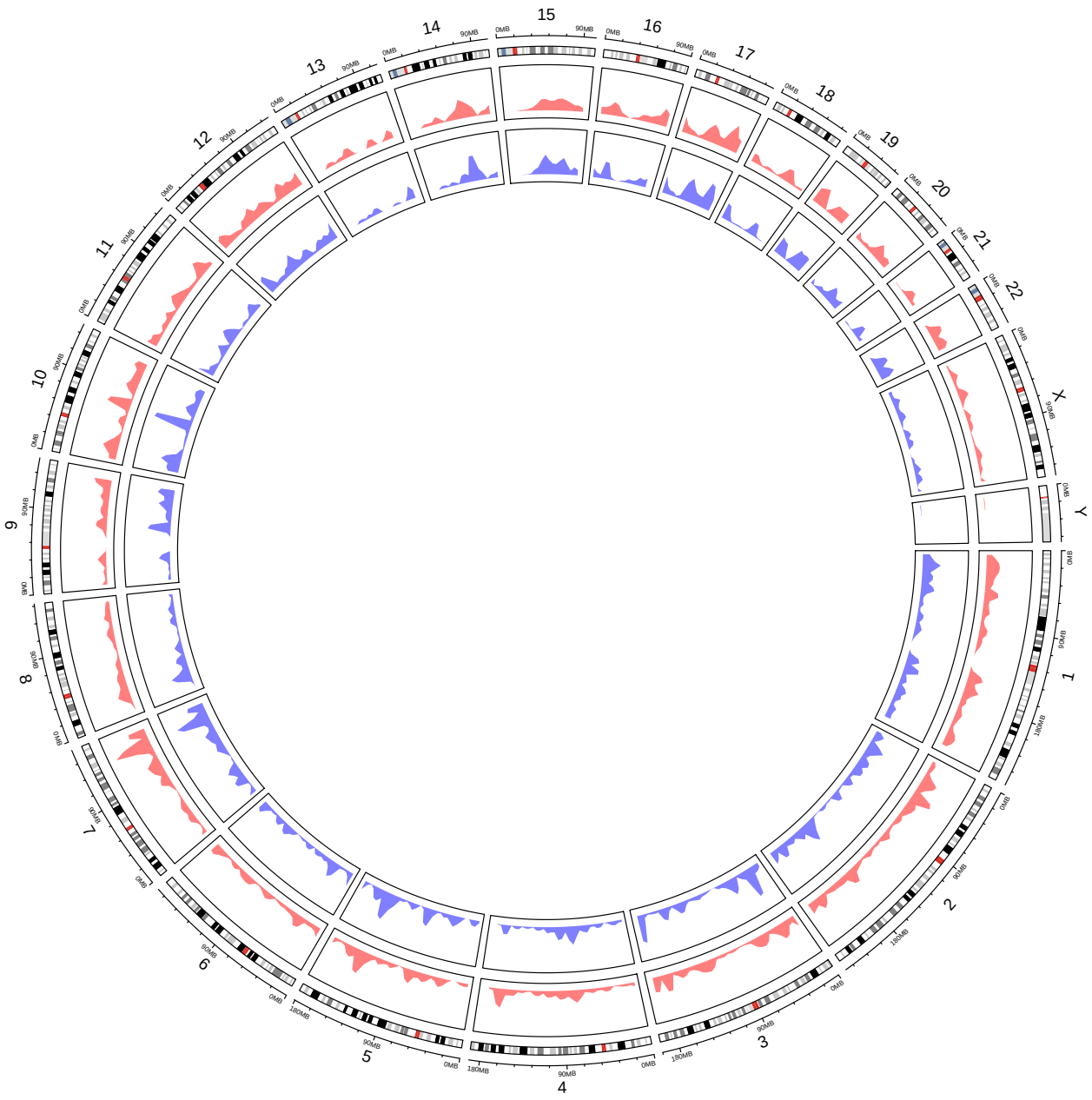
Computing % coverage

```
## compute coverage of query (peptide evidences) on subject (transcripts) by crhomosome
data("protein_coding_transcript_hg38") # load protein coding transcript as GRanges object

coverage <- computeCoverageByChromosome(query = granges_peptide,
                                        subject = transcripts_hg38,
                                        colName = 'Coverage')

coverage_mod <- computeCoverageByChromosome(query = granges_mod_peptide,
                                            subject = transcripts_hg38,
                                            colName = 'Coverage_mod')

# merging dfs
merged_coverage <- merge.data.frame(coverage, coverage_mod, by = 'Chromosome')

# ordering by chromosome
merged_coverage <- orderByChromosome(df = merged_coverage, colName = 'Chromosome')

print(merged_coverage)
```

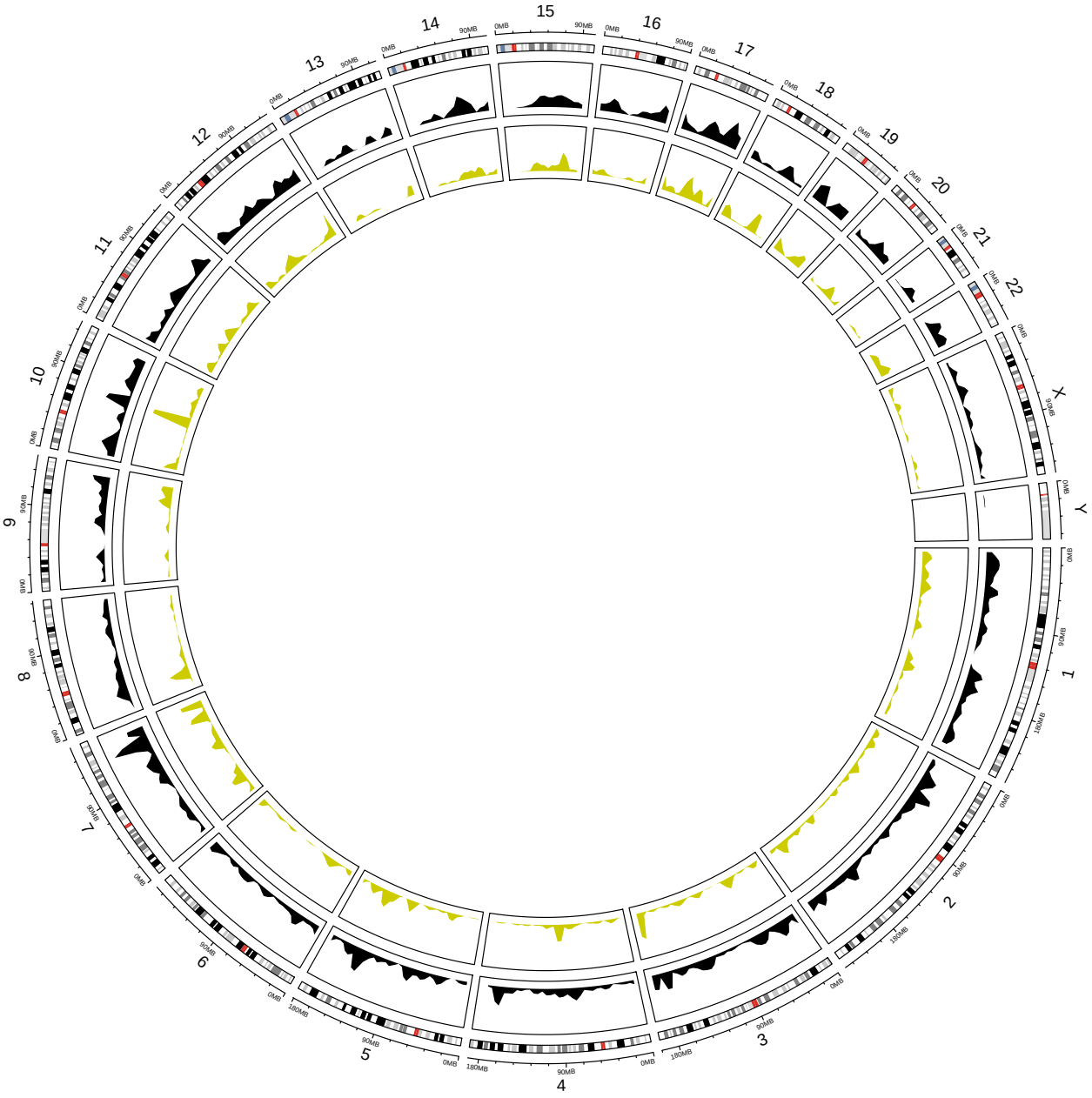
```
##    Chromosome Coverage Coverage_mod
## 1           1   50.781       50.049
## 12          2   50.613       49.873
## 16          3   46.526       46.556
## 17          4   51.239       50.553
## 18          5   47.049       46.096
## 19          6   51.290       49.855
## 20          7   58.682       56.440
## 21          8   52.876       52.573
## 22          9   48.021       47.601
## 2          10   44.897       44.374
## 3          11   46.371       45.456
## 4          12   43.799       43.021
## 5          13   31.403       31.038
## 6          14   49.515       48.434
## 7          15   46.962       47.143
## 8          16   47.596       47.133
## 9          17   51.952       51.845
## 10         18   51.426       51.050
## 11         19   48.749       48.536
## 13         20   49.875       49.560
## 14         21   34.799       35.251
## 15         22   47.774       47.045
## 24          X   42.346       42.574
## 25          Y   22.274       21.867
## 23          M    5.882        2.618
```

Visualizing the data

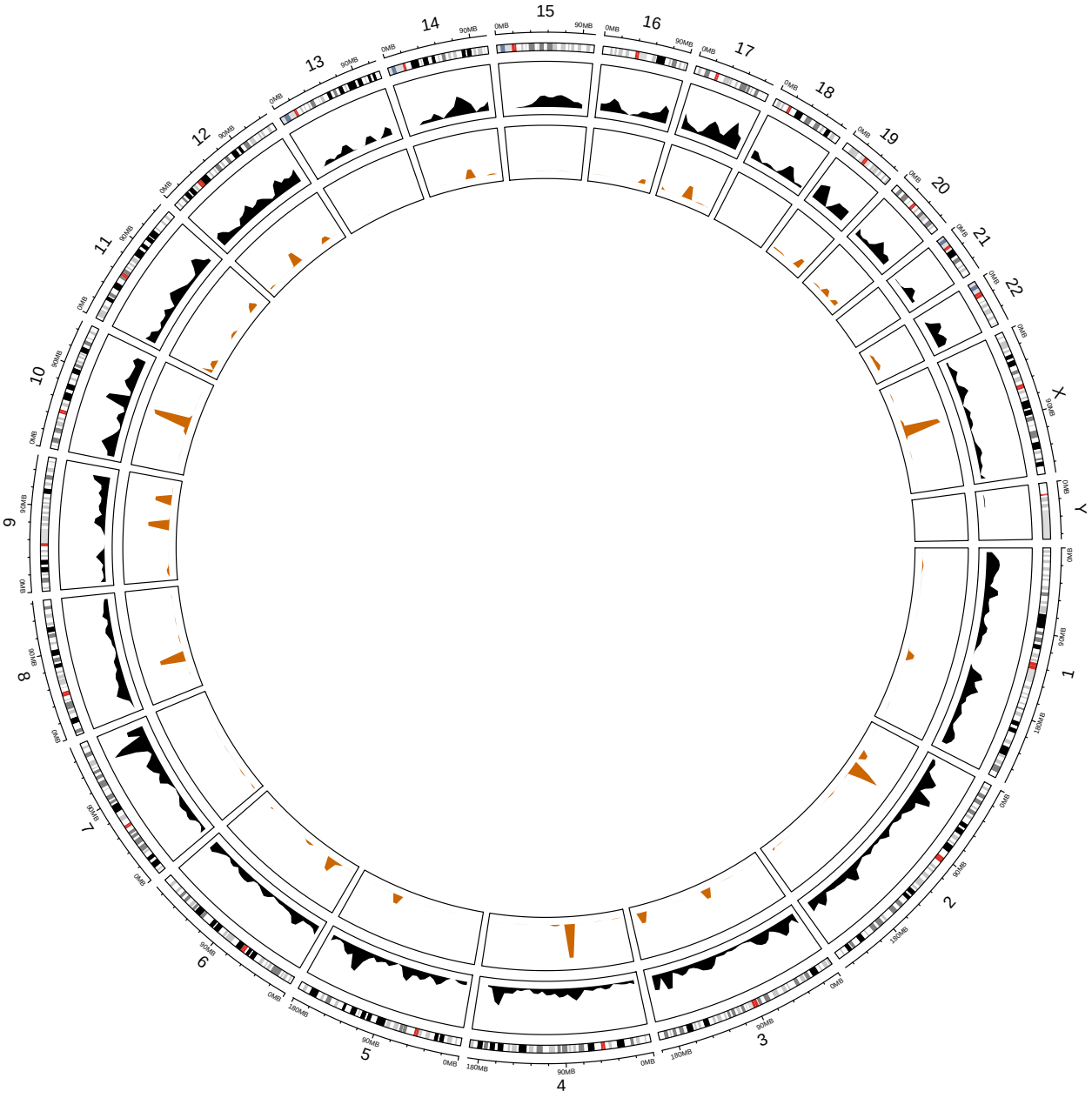
- The distribution of peptides by chromosome. (blue_track: modified peptide; red_track: non-modified)



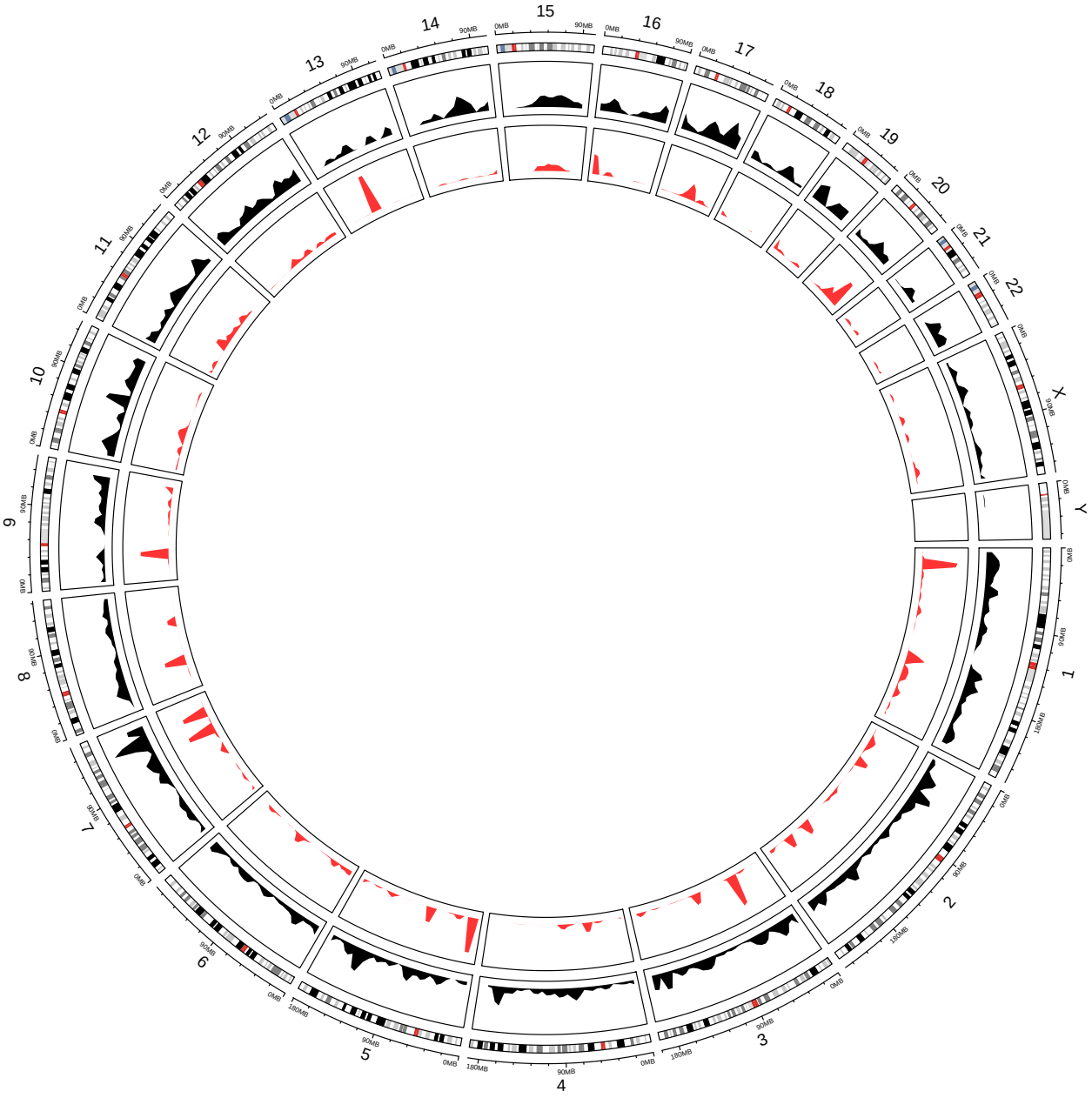
- The distribution of peptides by chromosome. (yellow_track: oxidation)



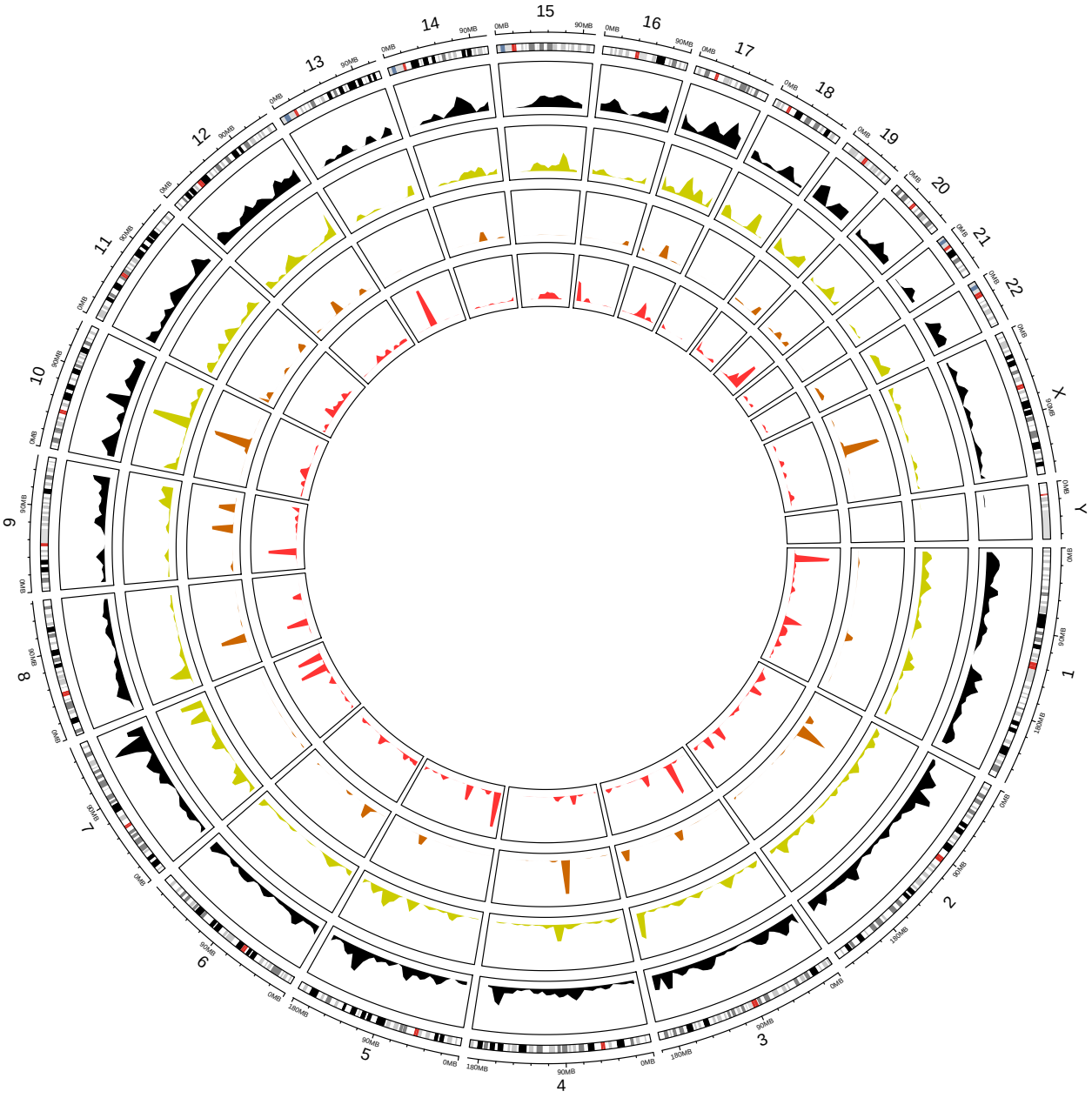
- The distribution of peptides by chromosome. (orange_track: acetyl)



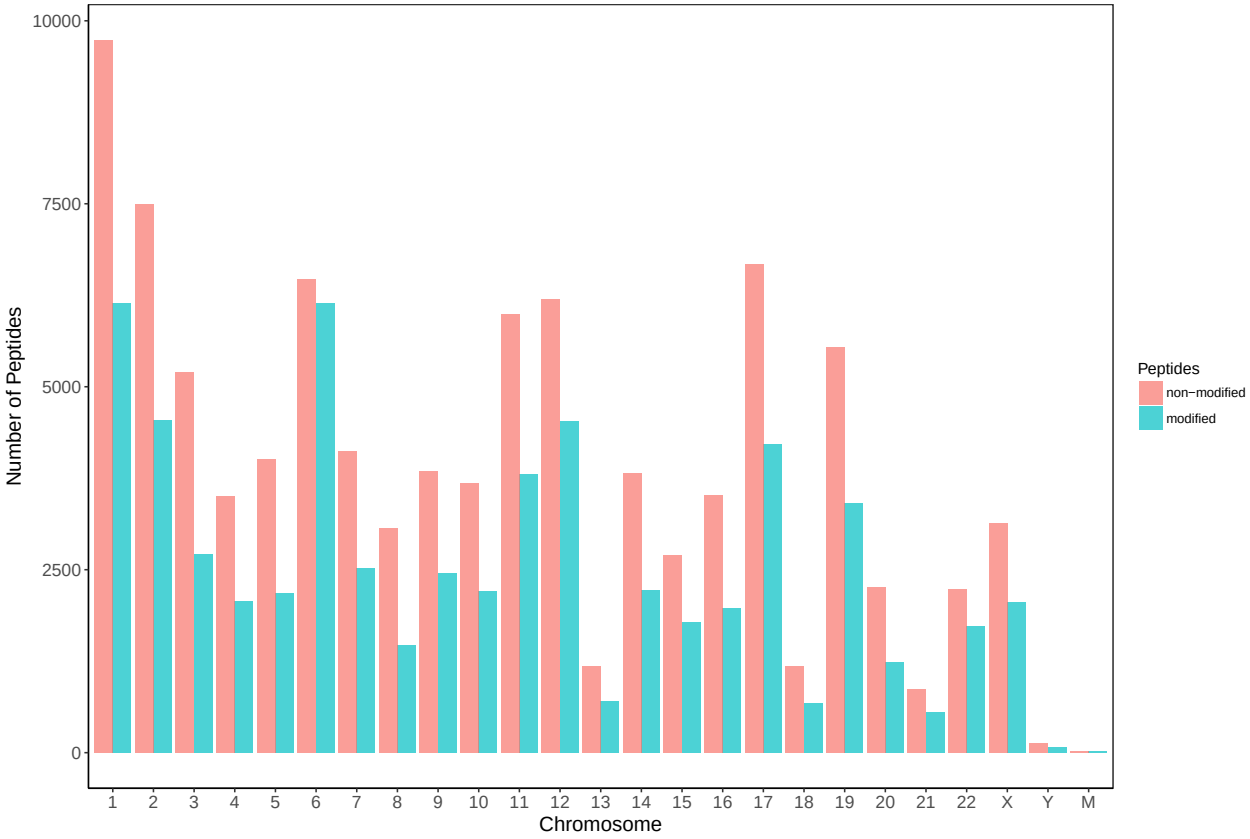
- The distribution of peptides by chromosome. (red_track: phospho)



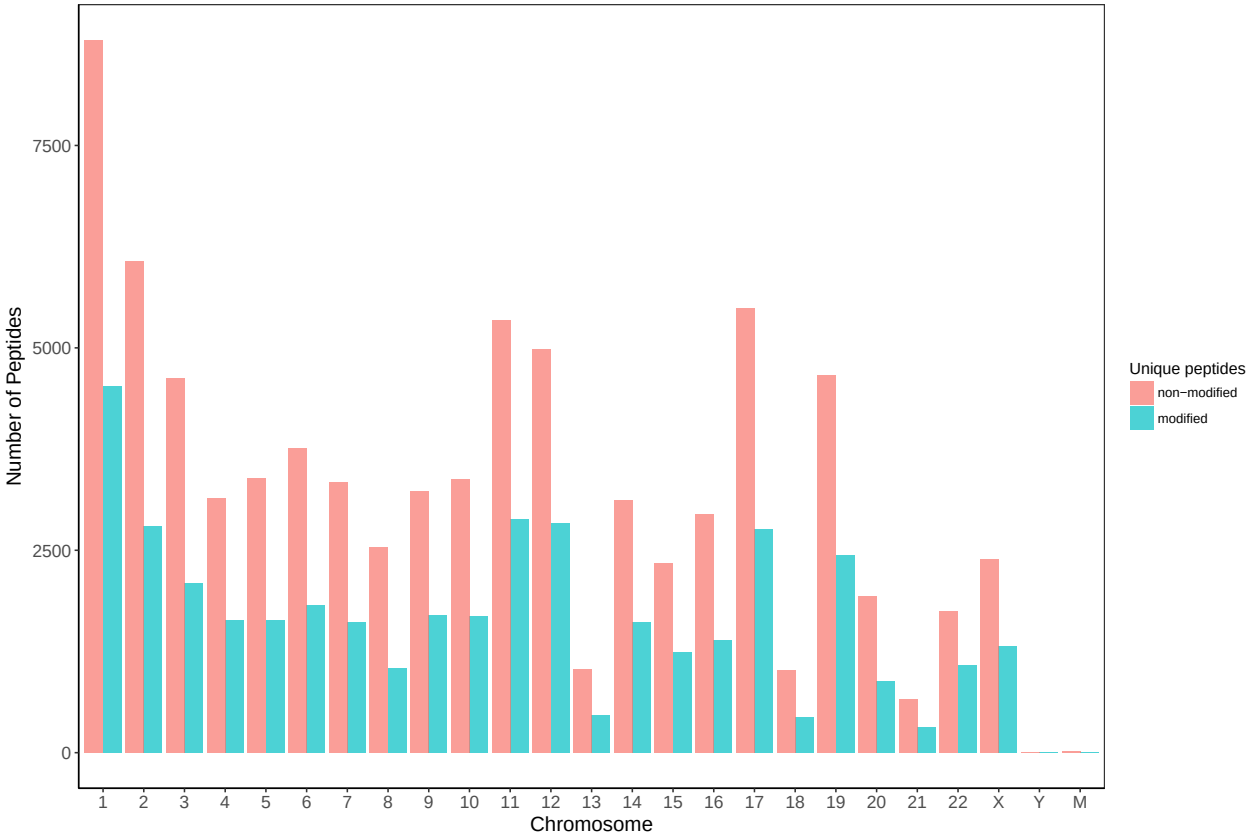
- The distribution of peptides by chromosome. (all peptides vs. oxidation, acetyl, phospho)



- barplot with number of peptides (modified and non-modified) by chromosome



- barplot with number of unique peptides (modified and non-modified) by chromosome



- barplot with coverage (all peptides) by chromosome