Genomic-Benchmarks Test

Table of contents

0.1 Libraries used

Python Code

```
# For genome-functions.R
library(stringr)
library(stringi)
library(primes)
# For parallel computing
library(doParallel)
library(foreach)
# For biological functions:
#   - Local/Global alignments
#   - DNA Shape computing
library(Biostrings)
library(DNAshapeR)
# For plotting
library(ggplot2)
library(dplyr)
library(plyr)
```

```
# For my own functions
source("/home/davidfm/Projects/UBMI-IFC/EnhaProm/scripts/genome-functions.R")
```

0.2 Downloading data

```
# Listing available datasets
from genomic_benchmarks.data_check import list_datasets
list_datasets()
# Inspecting each dataset to select two
from genomic_benchmarks.data_check import info as info_gb
info_gb("human_nontata_promoters") # <- This one will be used
info_gb("human_ensembl_regulatory")
info_gb("human_enhancers_cohn")    # <- This one will also be used
info_gb("human_enhancers_ensembl")
# Downloading datasets
from genomic_benchmarks.loc2seq import download_dataset
import os
os.chdir("/home/davidfm/Projects/UBMI-IFC/EnhaProm/datasets/GenomicBenchmarks")
download_dataset("human_nontata_promoters", version=0)
download_dataset("human_enhancers_cohn", version=0)
```

0.3 Formatting data

```
cd /home/davidfm/Projects/UBMI-IFC/EnhaProm/datasets/GenomicBenchmarks/
awk 'BEGIN{counter=0}{print ">promoter_"counter"|train|positive";
     print $0; counter+=1}' human_nontata_promoters/train/positive/*.txt \
     > promoters_train_positive.fasta
awk 'BEGIN{counter=0}{print ">enhancer_"counter"|train|positive";
     print $0; counter+=1}' human_enhancers_cohn/train/positive/*.txt \
     > enhancers_train_positive.fasta
```

0.4 Characterizing sequences

```
# Scanning sequences
prom_fasta <- "datasets/GenomicBenchmarks/promoters_train_positive.fasta"
enha_fasta <- "datasets/GenomicBenchmarks/enhancers_train_positive.fasta"
```

```r
prom_seqs <- scan(prom_fasta, character(), quote = "")[seq(2, 29484, 2)]
enha_seqs <- scan(enha_fasta, character(), quote = "")[seq(2, 20842, 2)]
# Prepairing clusters for parallel computing
corescluster <- makeCluster(6)
registerDoParallel(corescluster)
# Characterizing sequences and exporting to CSV
list_seqs <- list(promoters = prom_seqs, enhancers = enha_seqs)
reg_elems <- c("promoters", "enhancers")
for (reg_elem in reg_elems) {
  foreach(i = 1:6) %dopar% {
    library(stringr)
    library(stringi)
    library(primes)
    i_start <- ((i - 1) * 273) + 1
    i_final <- i * 273
    if (i > 1) {
      write.table(sequences_characterizer(list_seqs[[reg_elem]][i_start:i_final],
                                          k_max = 6, optim = TRUE),
                  paste("datasets/GB-Testing/test", reg_elem, "-minitraining_",
                        i, ".csv", sep = ""), sep = ",",
                  row.names = FALSE, col.names = FALSE)
    } else {
      write.csv(sequences_characterizer(list_seqs[[reg_elem]][i_start:i_final],
                                        k_max = 6, optim = TRUE),
                paste("datasets/GB-Testing/", reg_elem, "-minitraining_",
                      i, ".csv", sep = ""), row.names = FALSE)
    }
  }
}
```

0.5 Concatenating CSV’s

```bash
cat datasets/GB-Testing/testpromoters-minitraining_*.csv \
    > datasets/GB-Testing/test-1638-promoters-6mers.csv
cat datasets/GB-Testing/testenhancers-minitraining_*.csv \
    > datasets/GB-Testing/test-1638-enhancers-6mers.csv
```

0.6 Primary analysis

```
setwd("/home/davidfm/Projects/UBMI-IFC/EnhaProm")
testpromoters <- read.csv("datasets/GB-Testing/test-1638-promoters-6mers.csv",
                          check.names = F)
testenhancers <- read.csv("datasets/GB-Testing/test-1638-enhancers-6mers.csv",
                          check.names = F)
```

First we get an overviwew of the dimensions of our data:

```
dim(testpromoters)
```

```
[1]  1638 21830
```

```
dim(testenhancers)
```

```
[1]  1638 21830
```

It's noticeable the fact that we have way more columns than rows in this test table. Let's get a glimpse of the records corresponding to the first three promoters.

```
# knitr::kable(testpromoters[1:3,1:30])
```

A	T	C	G	temp	shan
0.1673307	0.2231076	0.3306773	0.2788845	87.21315	1.956136
0.2629482	0.2788845	0.2549801	0.2031873	81.00598	1.990374
0.3625498	0.2031873	0.2470120	0.1872510	80.02590	1.948719

k2.1_prod	k2.1_barc	k2.1_pals	k2.1_revc	k2.2_prod	k2.2_barc
9	1.959765	2.177403e+09	1.374020e+12	17.11198	1.531862
17	2.633165	1.138401e+17	2.971115e+17	26.44579	2.624612
38	5.347025	3.981015e+36	8.100763e+29	24.89016	1.855662

k2.2_pals	k2.2_revc	k2.3_prod	k2.3_barc	k2.3_pals	k2.3_revc
3.845422e+15	3.525451e+11	26.44579	2.445328	4.788062e+13	1.305836e+26
2.607331e+20	6.308689e+14	24.89016	2.513656	1.320117e+14	6.435389e+19
1.156904e+25	3.573059e+12	34.22397	3.431445	6.011592e+17	7.178542e+17

k2.3_revc	k2.4_prod	k2.4_barc	k2.4_pals	k2.4_revc	k2.5_prod	k2.5_barc
1.305836e+26	5.5	0.8121254	2.425939e+04	4.209683e+05	24.89016	2.163271
6.435389e+19	17.6	2.0647821	6.207541e+13	2.740751e+16	32.66833	2.801290
7.178542e+17	16.5	1.9121725	2.419251e+12	2.941608e+15	40.44651	3.119615

k2.5_pals	k2.5_revc	k2.6_prod	k2.6_barc	k2.6_pals	k2.6_revc
5.956114e+12	1.907086e+14	48	3.032966	1.321014e+23	8.039297e+19
9.988736e+16	2.535748e+18	34	2.389166	2.028406e+16	2.066402e+13
3.054415e+20	3.054391e+20	24	1.636697	5.764911e+11	8.636812e+10

```r
print(testenhancers[1:3,1:30])
```

```
      A     T     C     G    temp     shan k2.1_prod k2.1_barc    k2.1_pals
1 0.240 0.236 0.234 0.290 85.0392 1.993988        36 10.432345 1.885437e+37
2 0.204 0.286 0.210 0.300 84.4652 1.978249        20  6.153015 5.673403e+20
3 0.250 0.280 0.258 0.212 82.8252 1.992923        31  9.936447 1.640775e+32
     k2.1_revc k2.2_prod k2.2_barc    k2.2_pals    k2.2_revc k2.3_prod
1 8.632413e+30  24.89016  6.027803 1.579134e+29 1.707234e+31  76.22611
2 1.061673e+53  23.33452  4.984255 4.485619e+32 1.810188e+35  73.11484
3 5.563175e+39  46.66905  9.301359 9.951038e+37 3.612544e+25  60.66976
  k2.3_barc    k2.3_pals    k2.3_revc k2.4_prod k2.4_barc    k2.4_pals
1  13.43518 3.557032e+41 5.830130e+44      20.9  5.182017 2.534298e+18
2  13.29662 3.273426e+39 3.755909e+39      22.0  6.098819 2.555384e+17
3  12.53356 4.255600e+33 3.010280e+57      27.5  6.353834 3.581633e+31
     k2.4_revc k2.5_prod k2.5_barc    k2.5_pals    k2.5_revc k2.6_prod
1 2.395938e+20  43.55778  9.672032 1.852950e+24 3.825771e+48        80
2 2.343823e+21  48.22468 10.185028 1.559529e+27 2.552638e+46        58
3 2.035799e+26  57.55849 10.894642 3.730036e+31 2.060459e+35        60
  k2.6_barc    k2.6_pals    k2.6_revc
1 14.452432 1.791119e+41 4.233849e+41
2  7.732096 2.784552e+29 9.930851e+52
3  9.025510 1.288568e+31 2.183286e+27
```

```
meanpromoters <- colMeans(testpromoters)
meanenhancers <- colMeans(testenhancers)
sdpromoters <- apply(testpromoters, 2, sd)
sdenhancers <- apply(testenhancers, 2, sd)
# names_test <- rep(names(testpromoters),2)
cre_summary <- data.frame(Type = factor(rep(c("Promoter","Enhancer"),
                                         each = length(testpromoters))),
                          Field = rep(names(testpromoters),2),
                          Means = c(meanpromoters, meanenhancers),
                          StDevs = c(sdpromoters, sdenhancers))
# head(cre_summary)
knitr::kable(cre_summary[c(1:10,21831:21840),])
```

	Type	Field	Means	StDevs
1	Promoter	A	1.911256e-01	7.145510e-02
2	Promoter	T	1.995826e-01	7.599340e-02
3	Promoter	C	2.962655e-01	8.067840e-02
4	Promoter	G	3.130263e-01	8.524230e-02
5	Promoter	temp	8.720208e+01	5.379461e+00
6	Promoter	shan	1.891724e+00	9.437000e-02
7	Promoter	k2.1_prod	1.196276e+01	8.995306e+00
8	Promoter	k2.1_barc	1.500971e+00	1.229003e+00
9	Promoter	k2.1_pals	2.634405e+51	1.065029e+53
10	Promoter	k2.1_revc	9.789346e+62	3.960331e+64
21831	Enhancer	A	2.653712e-01	5.853000e-02
21832	Enhancer	T	2.678205e-01	5.815610e-02
21833	Enhancer	C	2.351111e-01	5.648440e-02
21834	Enhancer	G	2.316972e-01	5.435370e-02
21835	Enhancer	temp	8.269434e+01	3.785299e+00
21836	Enhancer	shan	1.959202e+00	3.892580e-02
21837	Enhancer	k2.1_prod	4.093346e+01	1.835697e+01
21838	Enhancer	k2.1_barc	1.208127e+01	5.751107e+00
21839	Enhancer	k2.1_pals	2.567318e+112	1.038419e+114
21840	Enhancer	k2.1_revc	3.321646e+121	1.344343e+123

```
# Get only 'prod' columns of each kmer
k2 <- n_ki(2); k3 <- n_ki(3); k4 <- n_ki(4); k5 <- n_ki(5); k6 <- n_ki(6)
kmer_sections_indexes <- c(1,
                           k2,
                           k2 + 1,
                            k2 + k3,
                            k2 + k3 + 1,
                            k2 + k3 + k4,
                            k2 + k3 + k4 + 1,
                            k2 + k3 + k4 + k5,
                            k2 + k3 + k4 + k5 + 1,
                            k2 + k3 + k4 + k5 + k6)
prod_indexes <- seq(7,21827,4)
barc_indexes <- seq(8,21828,4)
pals_indexes <- seq(9,21829,4)
revc_indexes <- seq(10,21830,4)
all_prod_indexes <- c(prod_indexes, 21830 + prod_indexes)
all_barc_indexes <- c(barc_indexes, 21830 + barc_indexes)
all_pals_indexes <- c(pals_indexes, 21830 + pals_indexes)
all_revc_indexes <- c(revc_indexes, 21830 + revc_indexes)
knitr::kable(head(testenhancers[, barc_indexes])[,c(1:8,length(barc_indexes))])
```

k2.1__barc	k2.2__barc	k2.3__barc	k2.4__barc	k2.5__barc	k2.6__barc	k2.7__barc	k2.8__barc	k6.4096__barc
10.432345	6.027803	13.435177	5.182017	9.672032	14.452432	2.0213095	13.731694	0.0000000
6.153015	4.984254	13.296620	6.098819	10.185028	7.732096	1.3635487	10.150355	0.3058662
9.936447	9.301359	12.533556	6.353834	10.894642	9.025510	1.3565393	16.335286	0.0000000
23.252879	8.744180	13.868229	9.853451	12.204317	3.958776	0.0930320	8.444184	0.0000000
3.814939	7.617881	9.290377	5.365261	10.457174	13.017956	1.4893074	14.727846	2.7938687
14.929810	6.939986	12.525913	11.864119	10.686542	2.227940	0.9421584	7.264738	0.0000000

```
knitr::kable(head(testenhancers[, barc_indexes], 5)[kmer_sections_indexes], table.attr = "qua
  kableExtra::kable_styling(full_width = FALSE) |>
    kableExtra::column_spec(column = 2:4, width = "0.4in")
```

k2.1__barc	k2.16__barc	k3.1__barc	k3.64__barc	k4.1__barc	k4.256__barc	k5.1__barc	k5.1024__barc	k6.1__barc	k6
10.432345	7.337222	3.4298932	5.133496	0.8724755	0.2043103	0.000000	0.0000000	0	
6.153015	19.441056	6.9076157	7.449599	0.2736402	1.8634239	0.000000	0.6130383	0	
9.936447	12.059759	9.6860719	4.791785	1.4367855	2.0880247	0.000000	0.8123330	0	
23.252879	12.044970	11.452795	4.043369	5.8892630	2.0034714	2.332278	0.4316400	0	
3.814939	11.720066	6.8633094	4.455931	0.0810805	3.7301540	0.000000	3.2607779	0	

```
knitr::kable(cre_summary[c(prod_indexes[1:5],(21830+prod_indexes)[1:5]),])
```

	Type	Field	Means	StDevs
7	Promoter	k2.1_prod	11.962760	8.995306
11	Promoter	k2.2_prod	15.840314	6.536126
15	Promoter	k2.3_prod	27.649084	9.205828
19	Promoter	k2.4_prod	8.638156	6.974283
23	Promoter	k2.5_prod	22.012519	7.995609
21837	Enhancer	k2.1_prod	40.933455	18.356973
21841	Enhancer	k2.2_prod	39.367630	9.765868
21845	Enhancer	k2.3_prod	58.095082	13.575118
21849	Enhancer	k2.4_prod	31.738828	12.985948
21853	Enhancer	k2.5_prod	57.394191	11.806409

```r
subset_cre_prod <- cre_summary[c(prod_indexes[17:64],(21830+prod_indexes)[17:64]),]
subset_cre_barc <- cre_summary[c(barc_indexes[17:64],(21830+barc_indexes)[17:64]),]
subset_cre_pals <- cre_summary[c(pals_indexes[17:64],(21830+pals_indexes)[17:64]),]
subset_cre_revc <- cre_summary[c(revc_indexes[17:64],(21830+revc_indexes)[17:64]),]
knitr::kable(cbind(subset_cre_prod[subset_cre_prod$Type=="Promoter",],
    subset_cre_prod[subset_cre_prod$Type=="Enhancer",])[,c(2,1,3,4,5,7,8)][1:10,])
```

	Field	Type	Means	StDevs	Type.1	Means.1	StDevs.1
71	k3.1_prod	Promoter	3.733822	4.441567	Enhancer	15.169109	10.176757
75	k3.2_prod	Promoter	3.208713	2.679694	Enhancer	9.520357	4.701251
79	k3.3_prod	Promoter	5.069128	3.448887	Enhancer	13.390625	5.694958
83	k3.4_prod	Promoter	2.363513	2.747660	Enhancer	9.863853	5.715001
87	k3.5_prod	Promoter	3.646951	3.485361	Enhancer	12.881026	5.906393
91	k3.6_prod	Promoter	5.124804	3.324229	Enhancer	11.042922	5.162449
95	k3.7_prod	Promoter	3.650813	3.084850	Enhancer	3.254059	3.143177
99	k3.8_prod	Promoter	3.756481	2.794403	Enhancer	11.358009	4.689413
103	k3.9_prod	Promoter	5.426771	4.453803	Enhancer	14.490419	6.240096
107	k3.10_prod	Promoter	9.318083	4.745578	Enhancer	15.674036	6.345508

```r
field_order <- subset_cre_prod$Field[1:48]
subset_cre_prod$Means
```

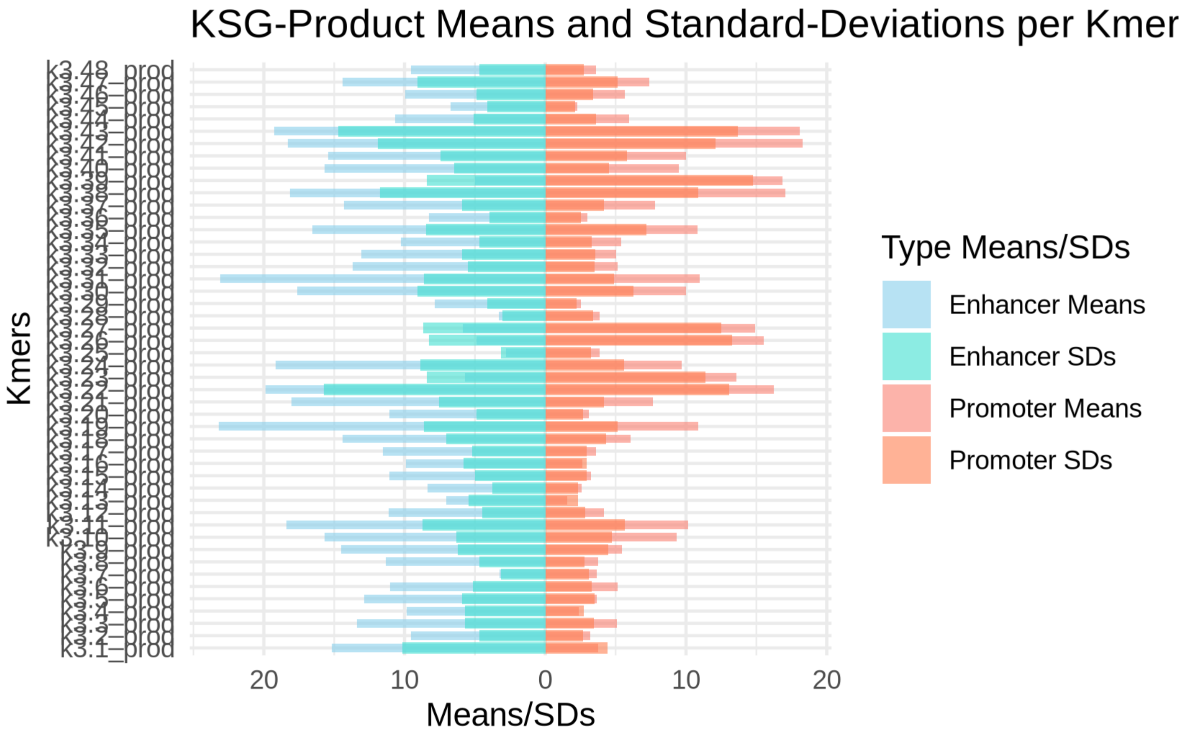
```
 [1]  3.733822  3.208713  5.069128  2.363513  3.646951  5.124804  3.650813
 [8]  3.756481  5.426771  9.318083 10.158642  4.176949  1.544579  2.586327
[15]  3.239394  2.656703  3.611691  6.037643 10.872411  3.075680  7.635905
[22] 16.233211 13.569377  9.680538  3.858207 15.529754 14.896730  3.878496
[29]  2.538018 10.020076 10.994143  5.127896  5.050658  5.404646 10.835603
```

```
[36]  2.981746  7.776152 17.063672 16.874431  9.484900  9.996806 18.264419
[43] 18.109890  5.977352  2.278580  5.645854  7.411662  3.606654 15.169109
[50]  9.520357 13.390625  9.863853 12.881026 11.042922  3.254059 11.358009
[57] 14.490419 15.674036 18.408043 11.153142  7.060554  8.359265 11.077995
[64]  9.899835 11.545321 14.397051 23.188692 11.087836 18.015617 19.891331
[71]  5.727871 19.145295  2.791931  4.866958  5.859807  3.292382  7.877964
[78] 17.602037 23.114300 13.710489 13.085873 10.235576 16.551690  8.261753
[85] 14.303432 18.128486  5.008222 15.694325 15.411434 18.289740 19.267399
[92] 10.688575  6.708704  9.933502 14.408686  9.522875
```

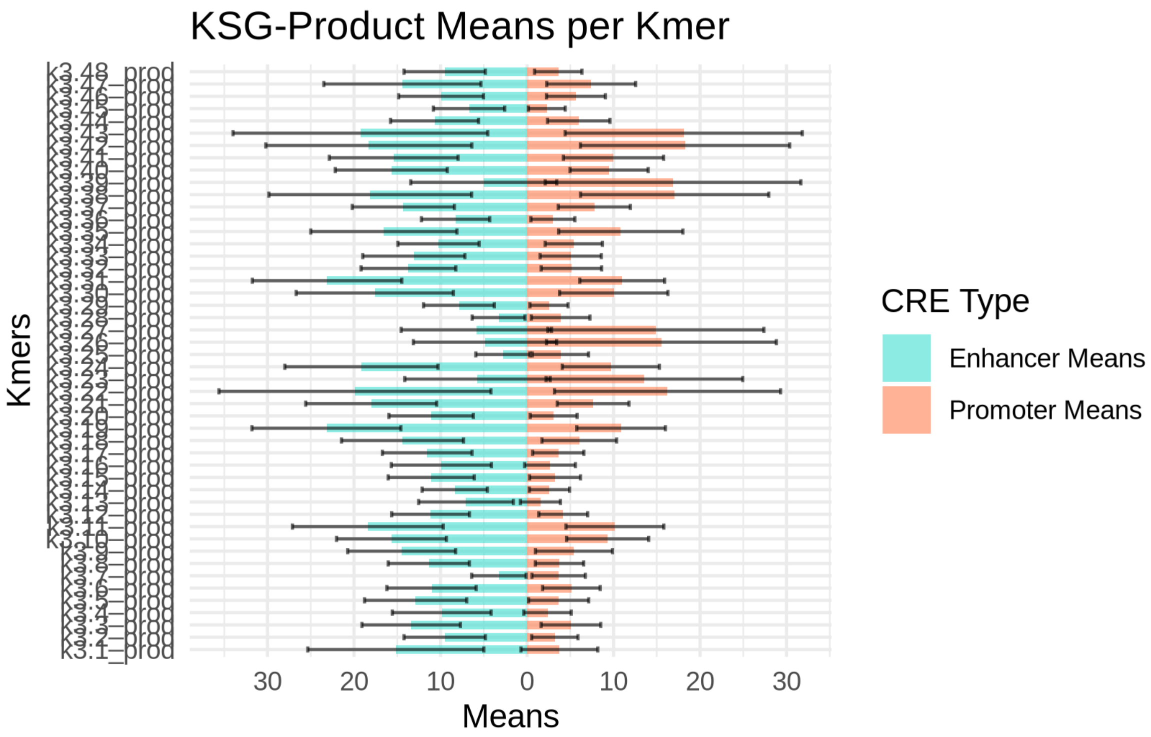
```
subset_cre_prod$StDevs
```

```
 [1]  4.441567  2.679694  3.448887  2.747660  3.485361  3.324229  3.084850
 [8]  2.794403  4.453803  4.745578  5.654427  2.825089  2.320564  2.326571
[15]  2.940212  2.932436  2.961735  4.319168  5.122931  2.710304  4.148137
[22] 13.073051 11.370095  5.614338  3.258066 13.292805 12.485377  3.391027
[29]  2.205837  6.267927  4.903813  3.502554  3.538477  3.306454  7.179443
[36]  2.542758  4.163639 10.888245 14.776918  4.521492  5.794870 12.101920
[43] 13.708462  3.609159  2.133656  3.406684  5.143815  2.739559 10.176757
[50]  4.701251  5.694958  5.715001  5.906393  5.162449  3.143177  4.689413
[57]  6.240096  6.345507  8.713120  4.492683  5.473509  3.767892  4.972782
[64]  5.789404  5.183821  7.055220  8.610000  4.881815  7.569853 15.722251
[71]  8.408784  8.852986  3.128115  8.279593  8.690941  3.054289  4.096377
[78]  9.084188  8.638399  5.479771  5.904495  4.697275  8.449506  3.947652
[85]  5.899648 11.718258  8.439162  6.478037  7.443246 11.906084 14.723036
[92]  5.092138  4.127505  4.906655  9.083055  4.695775
```

```
ggplot(subset_cre_prod) +
  geom_bar(aes(x = factor(Field, levels = field_order),
               y = ifelse(Type == "Enhancer", -Means, Means),
                          fill = paste(Type, "Means")),
               stat = "identity", position = "identity",
               alpha = 0.6, width = 0.7) +
  geom_bar(aes(x = factor(Field, levels = field_order),
               y = ifelse(Type == "Enhancer", -StDevs, StDevs),
                          fill = paste(Type, "SDs")),
               stat = "identity", position = "identity",
               alpha = 0.6) +
  coord_flip() +
  scale_y_continuous(breaks=seq(-30, 30, 10), labels=abs(seq(-30, 30, 10))) +
  scale_fill_manual(values = c("Enhancer Means" = "skyblue",
                               "Promoter Means" = "salmon",
                               "Enhancer SDs" = "turquoise",
                               "Promoter SDs" = "coral")) +
  labs(y = "Means/SDs", x = "Kmers",
       title = "KSG-Product Means and Standard-Deviations per Kmer",
       fill = "Type Means/SDs") +
  theme_minimal()
```



```
ggplot(subset_cre_prod) +
  geom_bar(aes(x = factor(Field, levels = field_order),
               y = ifelse(Type == "Enhancer", -Means, Means),
                          fill = paste(Type, "Means")),
               stat = "identity", position = "identity",
               alpha = 0.6, width = 0.7) +
  geom_errorbar(aes(x = factor(Field, levels = field_order),
                    ymin = ifelse(Type == "Enhancer",
                                  -Means + StDevs, Means - StDevs),
                    ymax = ifelse(Type == "Enhancer",
                                  -Means - StDevs, Means + StDevs)),
                    width = 0.5, colour = "black", alpha = 0.6) +
  coord_flip() +
  scale_y_continuous(breaks=seq(-30, 30, 10), labels=abs(seq(-30, 30, 10))) +
  scale_fill_manual(values = c("turquoise", "coral")) +
  labs(y = "Means", x = "Kmers",
       title = "KSG-Product Means per Kmer",
       fill = "CRE Type") +
  theme_minimal()
```


KSG-Product Means per Kmer
Kmers
k3.48_prod
k3.47_prod
k3.46_prod
k3.45_prod
k3.44_prod
k3.43_prod
k3.42_prod
k3.41_prod
k3.40_prod
k3.39_prod
k3.38_prod
k3.37_prod
k3.36_prod
k3.35_prod
k3.34_prod
k3.33_prod
k3.32_prod
k3.31_prod
k3.30_prod
k3.29_prod
k3.28_prod
k3.27_prod
k3.26_prod
k3.25_prod
k3.24_prod
k3.23_prod
k3.22_prod
k3.21_prod
k3.20_prod
k3.19_prod
k3.18_prod
k3.17_prod
k3.16_prod
k3.15_prod
k3.14_prod
k3.13_prod
k3.12_prod
k3.11_prod
k3.10_prod
k3.9_prod
k3.8_prod
k3.7_prod
k3.6_prod
k3.5_prod
k3.4_prod
k3.3_prod
k3.2_prod
k3.1_prod
Means
CRE Type
Enhancer Means
Promoter Means