

Specificity and Methylation sensitivity analysis of CTCF(F1-F9) and HoxB13

Zheng Zuo

Sep 12, 2020

Table of Contents

Background and Introduction.....	1
Importing and preprocessing data.....	2
Building regular specificity model from unmethylated sites.....	2
Building Methylation Sensitivity model.....	3
Pairwise comparison of unmethylated sites and M.SssI treated sites.....	3
Building Methylation sensitivity model by regression of 4L+1 parameters all together	5
Building Methylation sensitivity model by separate evaluation of methylation effect on each binding site.....	5
Why separate regression over specificity and methylation data performs better than all-in-one regression?.....	6
4L+1 modeling with artificial data exhibits the intrinsic limitation of all-in-one regression strategy	8
Conclusions.....	11

Background and Introduction

The CTCF protein, consisting of 11 tandem C2H2-type zinc fingers, is known to be critical insulator for genome architecture in mammals. Its finger 3 to 7 has been shown to recognize underlying reference sequence and motif strongly. Also It was known that CTCF processes some methylation sensitivity, i.e., when some CpG dinucleotide gets methylated within the binding site, the affinity of CTCF to the methylated sites gets compromised. To systematically investigate the specificity and methylation sensitivity of mouse CTCF protein, I designed and synthesized the following dsDNA libraries, covering every single and adjacent double variants of the core binding sites recognized by F3-F7. Position 19 serves as the barcode position to indicate wheather each sequence is treated by M.SssI methylated beforehand or not (For M.SssI treatment, only CpG dinucleotide can get methylated).

	F7			F6			F5			F4			F3						
Reference	C	C	A	C	T	A	G	G	G	G	C	A	C	T	A	T	G	T	
R1.M.Sssl	N	N	N	N	T	A	G	G	G	G	C	A	C	T	A	T	G	T	
R1.Un	N	N	N	N	T	A	G	G	G	G	C	A	C	T	A	T	G	A	
R2.M.Sssl	C	C	A	N	N	N	N	G	G	G	C	A	C	T	A	T	G	T	
R2.Un	C	C	A	N	N	N	N	G	G	G	C	A	C	T	A	T	G	A	
R3.M.Sssl	C	C	A	C	T	A	N	N	N	N	G	C	A	C	T	A	T	G	T
R3.Un	C	C	A	C	T	A	N	N	N	N	G	C	A	C	T	A	T	G	A
R4.M.Sssl	C	C	A	C	T	A	G	G	G	N	N	N	N	C	T	A	T	G	T
R4.Un	C	C	A	C	T	A	G	G	G	N	N	N	N	C	T	A	T	G	A
R5.M.Sssl	C	C	A	C	T	A	G	G	G	G	C	N	N	N	N	T	G	T	
R5.Un	C	C	A	C	T	A	G	G	G	G	C	N	N	N	N	T	G	A	
	1			4			7				10			13		16		19	

Importing and preprocessing data

```
load("../data/CTCF.rda")
(CTCF <- CTCF %>%
  dplyr::mutate(`Bound/Unbound` = Bound/Unbound,
    Energy = -log(Bound/Unbound)))

## # A tibble: 2,538 x 6
##   Sequence      Property Bound Unbound `Bound/Unbound` Energy
##   <chr>         <chr>    <dbl>  <dbl>         <dbl>  <dbl>
## 1 CCACTAGGGGGCGCTG me      3714    243          15.3   -2.73
## 2 CCACTAGGGGGCGCTG un      3184    232          13.7   -2.62
## 3 CCACTAGGGGGCGCTC un      2232    182          12.3   -2.51
## 4 CCACTAGGGGGCGCAC un      1494    127          11.8   -2.47
## 5 CCACTAGGGGGCGCTC me      1711    148          11.6   -2.45
## 6 CCACTAGGGGGCGCTA me     14185   1236          11.5   -2.44
## 7 CCACTAGGGGGCGCAG me      2408    210          11.5   -2.44
## 8 CCACTAGGGGGCGCAG un      2179    191          11.4   -2.43
## 9 CCACTAGGGGGCGCAA un      1577    139          11.3   -2.43
## 10 CCACTAGGGGGCGCCG me      2565    230          11.2   -2.41
## # ... with 2,528 more rows
```

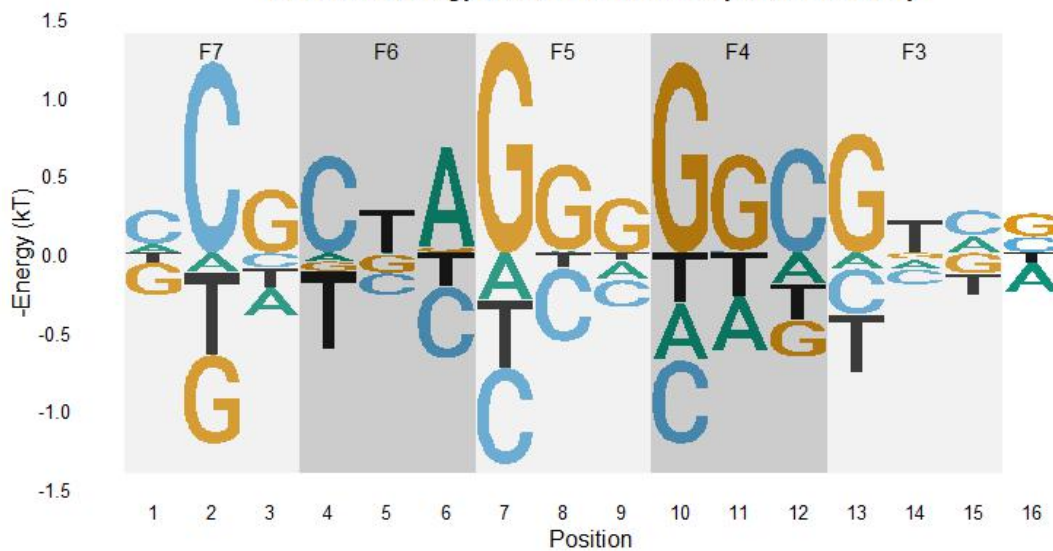
Building regular specificity model from unmethylated sites

It is easy to build specificity model and derive motif based on energy values from those unmethylated sites only.

```
CTCF.Model1.3Lp1<- CTCF %>%
  dplyr::filter(Property == "un") %>% # Select those untreated sites
  TFCookbook::buildEnergyModel(encoding = "3L+1")

CTCF.Model1.3Lp1%>%
  TFCookbook::getEnergyMatrix() %>%
  TFCookbook::plotEnergyLogo() +
  addFingers(index = 7:3, yMin = -1.4, yMax = 1.4) +
  labs(title = "Specificity profile of CTCF(F1-F9)\n based on energy values from unmethylated sites only") +
  theme(plot.title = element_text(hjust = 0.5))
```

Specificity profile of CTCF(F1-F9)
based on energy values from unmethylated sites only



```
#ggsave("Regular motif.svg", plot = last_plot(), height = 4.5, width = 9)
```

Building Methylation Sensitivity model

Pairwise comparison of unmethylated sites and M.SssI treated sites

Since the designed libraries cover both M.SssI treated and untreated sequences in one-to-one correspondence, it is possible to do pairwise comparison for each site. Note that M.SssI treatment doesn't necessarily mean methylation of cytosine, only those cytosines within CpG dinucleotide can get methylated, therefore those CpG-non-containing sequences can serve as negative control to gauge the intrinsic variances of our Methyl-Spec-seq measurement.

```
CTCF.pairwise <- inner_join(subset(CTCF, Property=="un"),
                             subset(CTCF, Property=="me"), by = "Sequence") %>%
  dplyr::select(Sequence,
                Energy.un = `Energy.x`,
                Energy.me = `Energy.y`) %>%
  dplyr::mutate(CpG.containing = as.integer(stringi::stri_count_fixed(Sequence, "CG")),
                Energy = Energy.me - Energy.un)
```

```
CTCF.pairwise %>%
  dplyr::arrange(desc(Energy)) %>%
  dplyr::filter(Energy.un <= 1.5) %>%
  dplyr::rename(`Number of CpG sites` = CpG.containing,
                 `Methylation effect (kT)` = Energy)
```

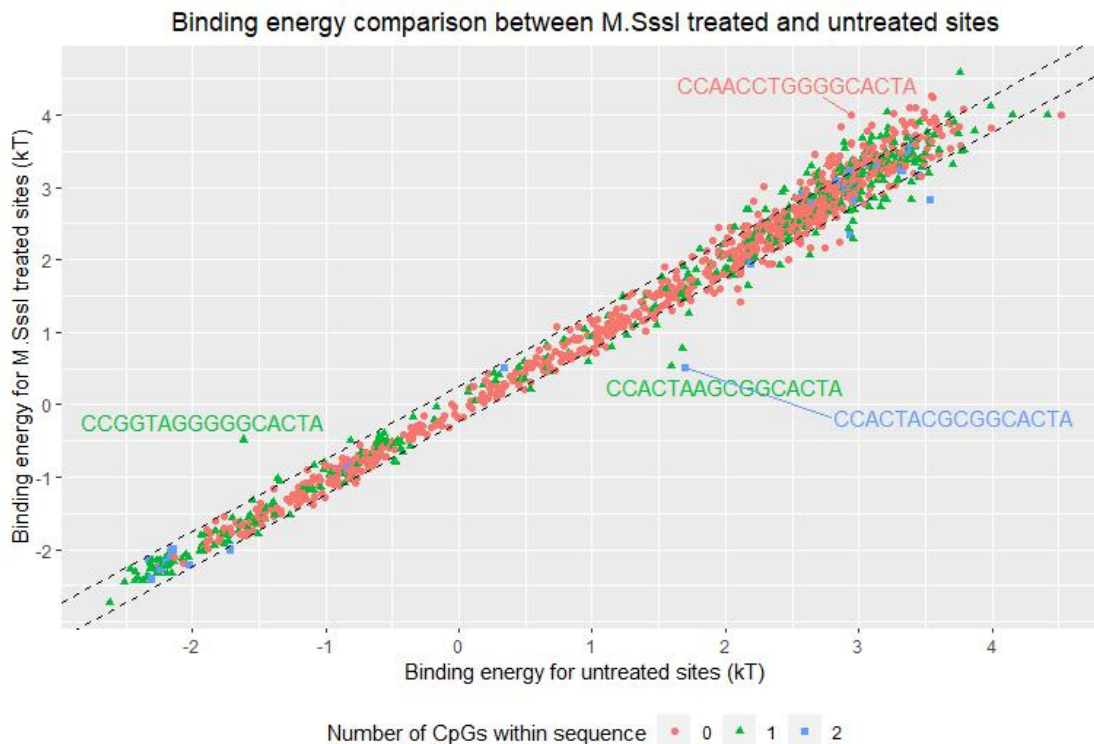
```
## # A tibble: 523 x 5
```

	Sequence	Energy.un	Energy.me	`Number of CpG sites`	`Methylation effect (kT)`
	<chr>	<dbl>	<dbl>	<int>	<dbl>
## 1	CCGGTAGGGGGCA~	-1.62	-0.488	1	1.13
## 2	TCGCTAGGGGGCA~	-1.36	-1.01	1	0.343
## 3	CTCCTAGGGGGCA~	0.727	1.07	0	0.343
## 4	ACGGTAGGGGGCA~	-0.814	-0.489	1	0.325
## 5	CCGATAGGGGGCA~	-1.34	-1.07	1	0.273
## 6	CCACTAGGGCCCG~	1.23	1.49	1	0.258
## 7	CCACTAGGGGGCC~	-1.14	-0.891	0	0.247
## 8	CCACGAGGGGGCA~	-1.08	-0.842	1	0.242

```
## 9 TTACTAGGGGGCA~      1.30      1.53      0      0.234
## 10 CCACTAGGGGGCC~     -1.39     -1.16      0      0.227
## # ... with 513 more rows
```

Clearly, for those high-affinity, or low-binding energy sequences, CCGGTAGGGGGGCACTA deviate significantly from the diagonal lines by up to $\sim 1\text{kT}$, whereas almost all non-CpG containing sites fall within the 0.25kT energy deviation bounds, so in our measurement mCpG at position 2 significantly compromise the binding affinity of CTCF, which is consistent with previous literature report. For those weak binding sites, there are certain degree of divergence, very likely because of the alternative recognition mode of CTCF including the nearby barcode position, thus we consider that could be technical artifact.

```
CTCF.pairwise %>%
  dplyr::mutate(Label = if_else(abs(Energy.un - Energy.me) > 1, Sequence, ""),
                CpG.containing = as.factor(CpG.containing)) %>%
  ggplot(aes(x = Energy.un, y = Energy.me, shape = CpG.containing, color = CpG.containing,
             label = Label)) +
  geom_point() +
  geom_abline(intercept = -0.25, slope = 1, linetype="dashed") +
  geom_abline(intercept = 0.25, slope = 1, linetype="dashed") +
  geom_text_repel(show.legend = FALSE, force = 10) +
  ggtitle("Binding energy comparison between M.SssI treated and untreated sites") +
  scale_x_continuous(breaks = seq(-2, 4, 1)) + scale_y_continuous(breaks = seq(-2, 4, 1))
+
  xlab("Binding energy for untreated sites (kT)") +
  ylab("Binding energy for M.SssI treated sites (kT)") +
  labs(shape = "Number of CpGs within sequence", color = "Number of CpGs within sequence")
+
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5))
```



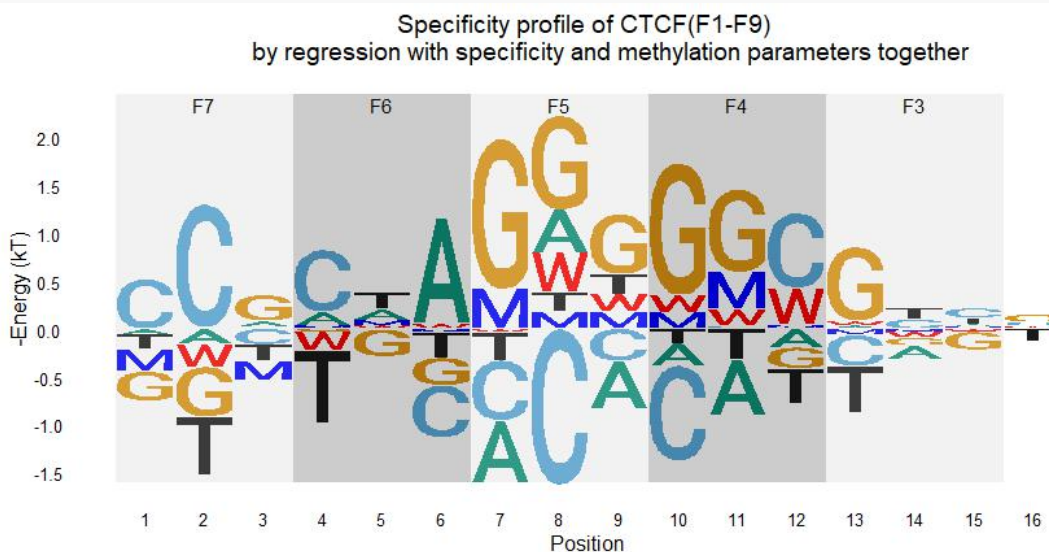
```
#ggsave("Pairwise comparison.svg", plot = last_plot(), height = 5.7, width = 8)
```

Building Methylation sensitivity model by regression of 4L+1 parameters all together

As discussed in the main text, if we use encoding scheme (3+1)L+1 and do regression analysis with all 4L+1 parameters altogether, we can get some methylation profiles shown below, which is very different from what we expect based on visual inspection, e.g., position 2 didn't show any negative methylation sensitivity at all and other positions show considerable amount of methyl effect.

```
CTCF.Model.4Lp1 <- CTCF %>%
  dplyr::mutate(Sequence = if_else(Property=="un",
                                   Sequence,
                                   stringi::stri_replace_all_fixed(Sequence, "CG", "MW")))
%>%
  dplyr::filter(Energy <= 1.7) %>%
  TFCookbook::buildEnergyModel(encoding = "4L+1")

CTCF.Model.4Lp1 %>%
  TFCookbook::getEnergyMatrix() %>%
  TFCookbook::addMethylMatrix(CTCF.Model.4Lp1, encoding = "(3+1)L+1") %>%
  TFCookbook::plotEnergyLogo() +
  addFingers(index = 7:3, -1.6, 2.45) +
  scale_y_continuous(breaks = seq(-1.5, 2, 0.5)) +
  labs(title = "Specificity profile of CTCF(F1-F9)
         by regression with specificity and methylation parameters together") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
#ggsave("4L+1 motif.svg", plot = last_plot(), height = 4.5, width = 9)
```

Building Methylation sensitivity model by separate evaluation of methylation effect on each binding site

Alternatively, if we perform regression over methylation-related parameters (1MW to 16MW) alone using pairwise comparison data, it is easy to derive a methylation sensitivity model that matches our visual inspection well. Note that our measurement resolution is around 0.25kT, so we can choose 0.25kT as cut-off to analyze those significant methylation effect alone.

```
(CTCF.MethylModel <- CTCF.pairwise %>%
  filter(Energy.un <= 1.5, Energy > 0.25) %>% ## Filtering out low affinity binding sites
  and select only significant observation
  mutate(Sequence = stringi::stri_replace_all_fixed(Sequence, "CG", "MW"))) %>%
```



```

arrange(desc(Energy)) %>%
TFCookbook::buildMethylationModel(encoding = "(3+1)L+1", withIntercept = FALSE))

##
## Call:
## lm(formula = Energy ~ . - Energy + 0, data = input)
##
## Coefficients:
##  `1MW`  `2MW`  `3MW`  `4MW`  `5MW`  `6MW`  `7MW`  `8MW`  `9MW`  `10MW`
##      NA  0.5174      NA      NA      NA      NA      NA      NA      NA      NA
##  `11MW` `12MW` `13MW` `14MW` `15MW` `16MW`
##      NA  0.2582      NA      NA      NA      NA

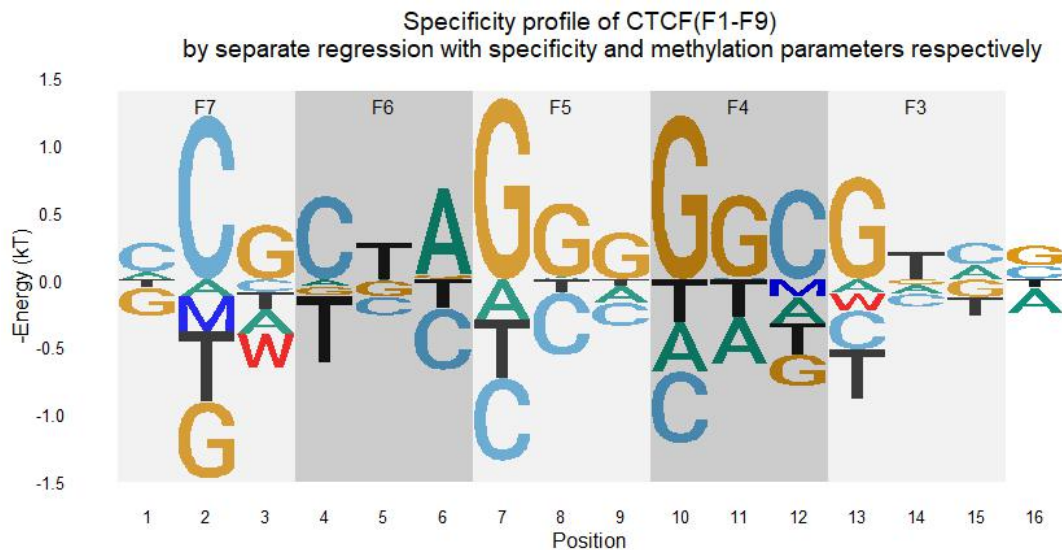
```

By combining specificity and methylation sensitivity info together, we can produce some motif logo that catches most of information to our expectation. Since we have no information about the strand-specific methylation effect in current experimental setup, we just plot it as 50/50 by default. It has been shown in other literature that the methylation effect at position (2,3) primarily comes from the upper strand.

```

CTCF.Model.3Lp1%>%
  TFCookbook::getEnergyMatrix() %>% ## Getting energy matrix for regular specificity model
  TFCookbook::addMethylMatrix(CTCF.MethylModel, encoding = "(3+1)L+1") %>% ## Adding methylation parameters to the matrix
  TFCookbook::plotEnergyLogo() +
  addFingers(index = 7:3, -1.5, 1.4) +
  scale_y_continuous(breaks = seq(-1.5, 1.5, 0.5)) +
  labs(title = "Specificity profile of CTCF(F1-F9)
        by separate regression with specificity and methylation parameters respectively")
+
  theme(plot.title = element_text(hjust = 0.5))

```



```

#ggsave("Separate regressions motif.svg", plot = last_plot(), height = 4.5, width = 9)

```

Why separate regression over specificity and methylation data performs better than all-in-one regression?

To illustrate the intrinsic limitation of all-in-one regression of all parameters, we can compare all those predicted binding energy for unmethylated and methylated sites at each position based on 4L+1 model with the observed values as following figure. Clearly, for those sequences with CpG at position 1 and 11, each pair exhibited almost no methylation effect, but we still get non-zero methylation parameters (1MW

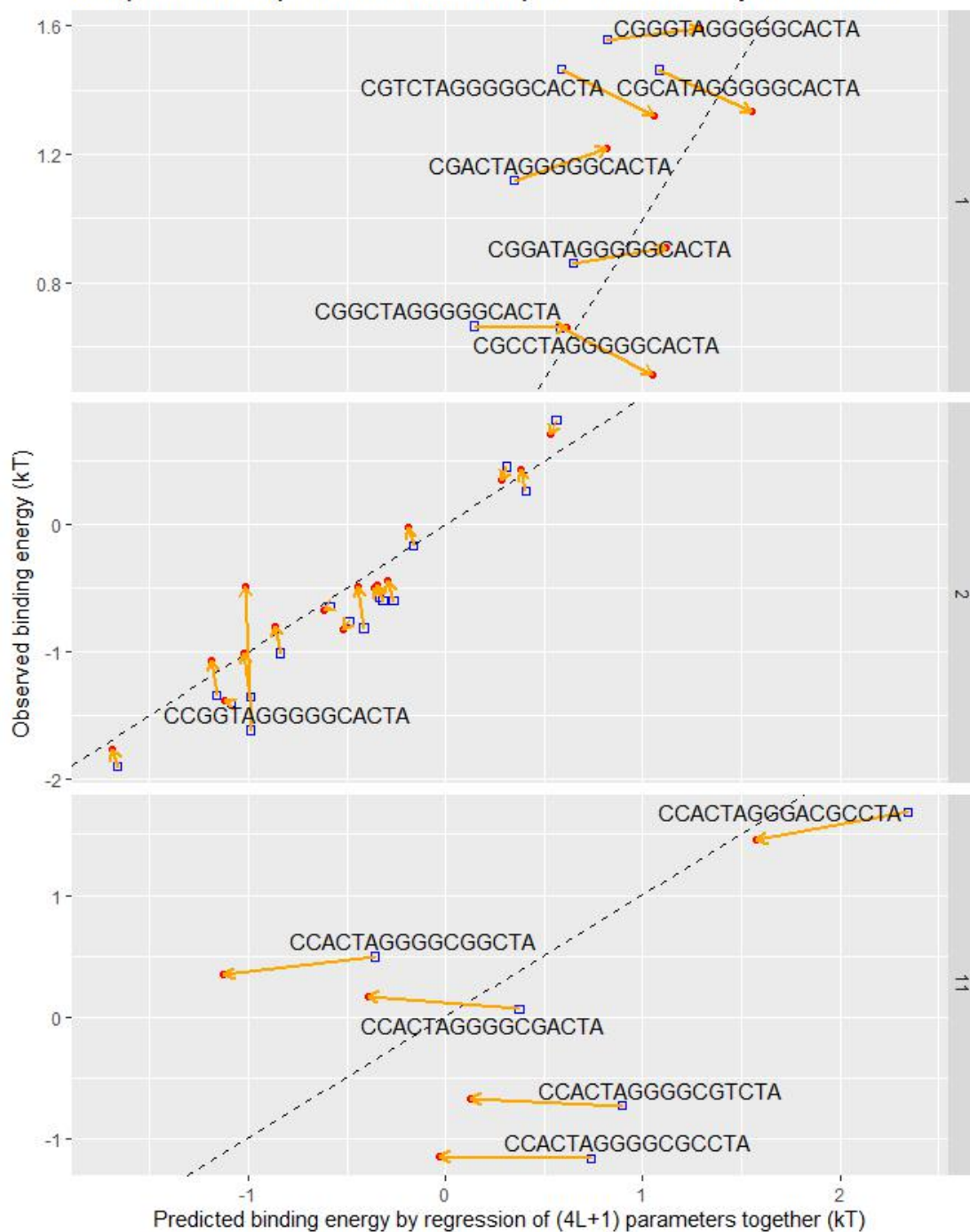
and 11MW), most likely because non-zero methyl- parameters decreases the overall deviation between observed and predicted values and “push” most pairs closer to the diagonal lines. On the other hand, for sequences with CpG at position 2, only one sequence (CCGGTAGGGGGCACTA) showed significant methylation effect, which gets “buried” in other insignificant ones, and thus we couldn’t get some positive 2MW parameter with 4L+1 regression method.

```
CTCF.4Lp1 <- CTCF %>%
  dplyr::mutate(position.CpG = stringi::stri_locate_first_fixed(Sequence, "CG")[,"start"])
  %>%
  dplyr::filter(Energy <= 1.7) %>%
  dplyr::mutate(Predicted.Energy = CTCF.Model.4Lp1$fitted.values) ##Adding predicted values by 4L+1 regression method

CTCF.4Lp1.paired <-
  dplyr::inner_join(subset(CTCF.4Lp1, Property == "un"),
                    subset(CTCF.4Lp1, Property == "me"),
                    by = "Sequence") %>%
  dplyr::select(Sequence,
                position.CpG = position.CpG.x,
                Energy.un = Energy.x,
                Predicted.Energy.un = Predicted.Energy.x,
                Energy.me = Energy.y,
                Predicted.Energy.me = Predicted.Energy.y)

CTCF.4Lp1.paired %>%
  dplyr::filter(position.CpG %in% c(1, 2, 11)) %>%
  ggplot() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  geom_point(aes(x = Predicted.Energy.un, y = Energy.un), color = "blue", shape = 0) +
  geom_point(aes(x = Predicted.Energy.me, y = Energy.me), color = "red", shape = 19) +
  geom_segment(aes(x = Predicted.Energy.un, y = Energy.un,
                  xend = Predicted.Energy.me, yend = Energy.me),
              lineend = 'round', linejoin = 'bevel', size = 1, color = "orange",
              arrow = arrow(length = unit(0.03, "npc"))) +
  geom_text_repel(data = function(x) filter(x, ((position.CpG != 2) | (Sequence == "CCGGTAGGGGGCACTA"))),
                  aes(x = Predicted.Energy.un, y = Energy.un, label = Sequence), force
= 1) +
  ggtitle("Comparison of experimental data with predicted values by 4L+1 model") +
  xlab("Predicted binding energy by regression of (4L+1) parameters together (kT)") +
  ylab("Observed binding energy (kT)") +
  facet_wrap(~position.CpG,
            scales = "free_y",
            strip.position = "right",
            nrow = 3)
```

Comparison of experimental data with predicted values by 4L+1 model



```
#ggsave("Model 4Lp1 comparison.svg", plot = last_plot(), height = 9, width = 7)
```

4L+1 modeling with artificial data exhibits the intrinsic limitation of all-in-one regression strategy

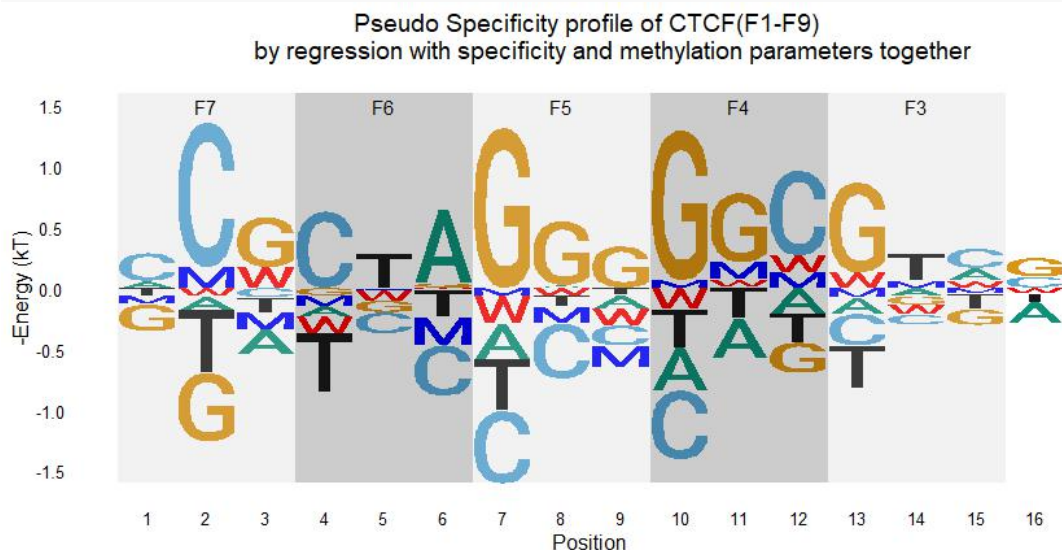
Besides comparison of experimental data with predicted values by 4L+1 model, we can create an artificial “pseudo” dataset in which each methylated site has exactly the same value as the unmethylated one, and construct 4L+1 model to illustrate the intrinsic limitation of this all-in-one regression strategy.

```
CTCF.pseudoMe <- subset(CTCF, Property == "un") %>%
  dplyr::mutate(Sequence = stringi::stri_replace_all_fixed(Sequence, "CG", "MW"),
    Property = "me") %>%
  dplyr::filter(stringr::str_detect(Sequence, "MW"))
```



```
CTCF.pseudoModel.4Lp1 <- rbind(subset(CTCF, Property == "un"), CTCF.pseudoMe) %>%
  TFCookbook::buildEnergyModel(encoding = "4L+1")

CTCF.pseudoModel.4Lp1 %>%
  TFCookbook::getEnergyMatrix() %>%
  TFCookbook::addMethylMatrix(MethylModel = CTCF.pseudoModel.4Lp1, encoding = "(3+1)L+1")
  %>%
  TFCookbook::plotEnergyLogo() +
  addFingers(index = 7:3, -1.6, 1.6) +
  scale_y_continuous(breaks = seq(-1.5, 1.5, 0.5)) +
  labs(title = "Pseudo Specificity profile of CTCF(F1-F9)
    by regression with specificity and methylation parameters together") +
  theme(plot.title = element_text(hjust = 0.5))
```



Clearly, under this “absolute no methylation effect” scenario, we still derive some non-zero methylation parameters from 4L+1 model, which shows us it is more realistic to build separate methylation effect model based on pairwise comparison between each individual site.

```
CTCF.pseudo.paired <- subset(CTCF, Property == "un") %>%
  dplyr::mutate(Sequence.MW = stringi::stri_replace_all_fixed(Sequence, "CG", "MW"),
    position.CpG = stringi::stri_locate_first_fixed(Sequence, "CG")[, "start"],
    Predicted.Energy.un = TFCookbook::predictEnergyMW(Sequence, CTCF.pseudoModel.4Lp1),
    Predicted.Energy.me = TFCookbook::predictEnergyMW(Sequence.MW, CTCF.pseudoModel.4Lp1)) %>%
  dplyr::select(Sequence, Sequence.MW, position.CpG, Energy, Predicted.Energy.un, Predicted.Energy.me)

## Warning: Expected 64 pieces. Additional pieces discarded in 1266 rows [1, 2, 3,
## 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

## Warning in predict.lm(model, newdata = ., type = "response"): prediction from a
## rank-deficient fit may be misleading

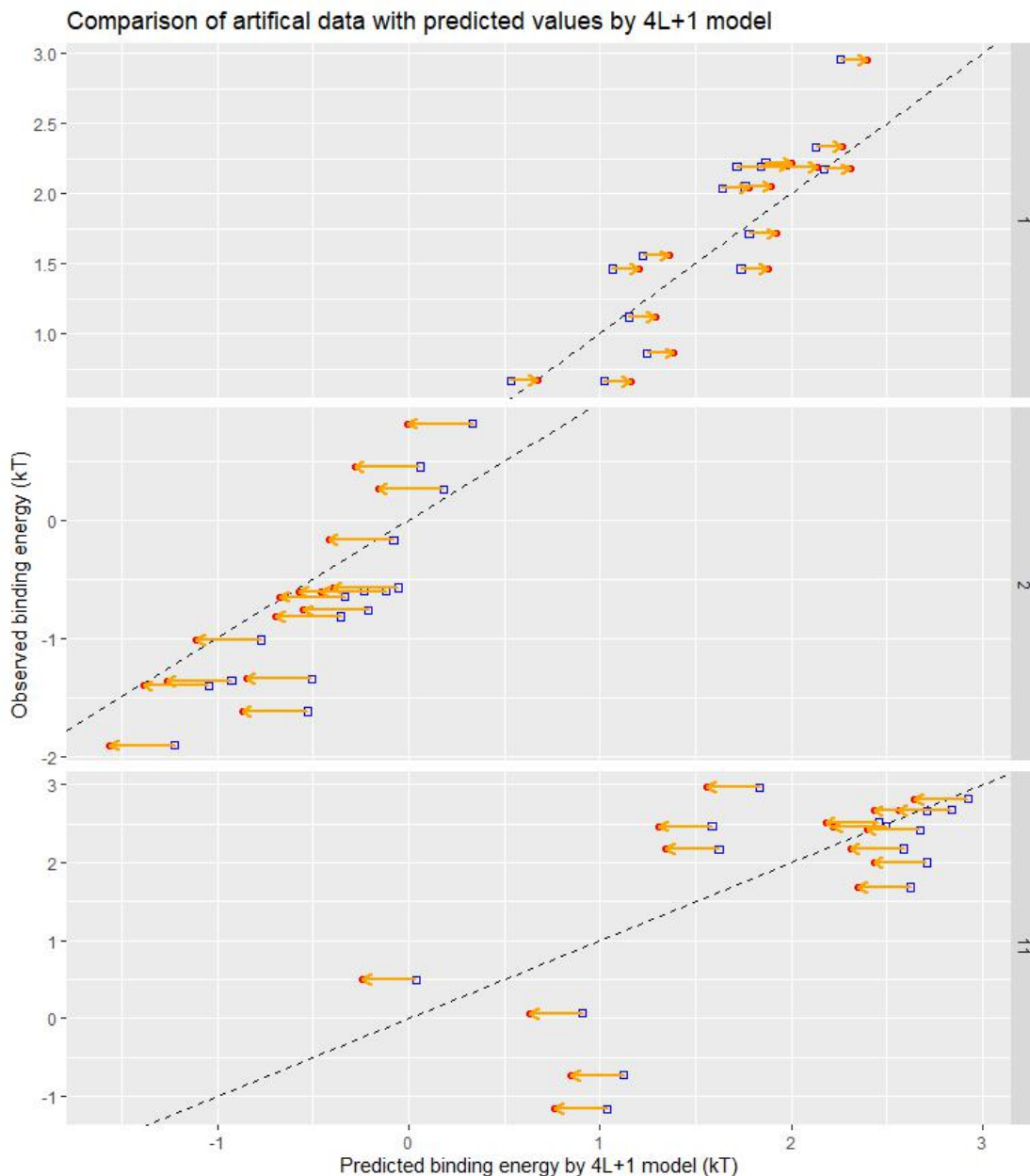
## Warning: Expected 64 pieces. Additional pieces discarded in 1266 rows [1, 2, 3,
## 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

## Warning in predict.lm(model, newdata = ., type = "response"): prediction from a
## rank-deficient fit may be misleading
```

```

CTCF.pseudo.paired %>%
  dplyr::filter(position.CpG %in% c(1, 2, 11)) %>%
  ggplot() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  geom_point(aes(x = Predicted.Energy.un, y = Energy), color = "blue", shape = 0) +
  geom_point(aes(x = Predicted.Energy.me, y = Energy), color = "red", shape = 19) +
  geom_segment(aes(x = Predicted.Energy.un, y = Energy,
                  xend = Predicted.Energy.me, yend = Energy),
              lineend = 'round', linejoin = 'bevel', size = 1, color = "orange",
              arrow = arrow(length = unit(0.03, "npc")))) +
  ## geom_text_repel(aes(x = Predicted.Energy.un, y = Energy, Label = Sequence)) +
  ggtitle("Comparison of artificial data with predicted values by 4L+1 model") +
  xlab("Predicted binding energy by 4L+1 model (kT)") +
  ylab("Observed binding energy (kT)") +
  facet_wrap(~position.CpG,
            scales = "free_y",
            strip.position = "right",
            nrow = 3)

```



```
#ggsave("Model 4Lp1 comparison.eps", plot = last_plot(), height = 9, width = 8)
```

Conclusions

Based on above analysis, it is clear that:

- 1) It is unrealistic to use “all-in-one” regression strategy to construct specificity and methylation effect model for CTCF and potential many other TFs;
- 2) We can build a composite specificity and methylation effect model for CTCF by doing regression over methylation-irrelevant and methylation parameters separately. Since the measurement resolution of typical Spec-seq experiment is around 0.2 kT, we can perform pairwise comparison on each pair of sequences and drop all those insignificant results below certain threshold ($\sim 0.25\text{kT}$) and highlight those significant positions only.

Specificity and methylation sensitivity analysis of HoxB13

Zheng Zuo

Sep 16, 2020

Table of Contents

Introduction.....	11
Importing and preprocessing data.....	11
Building specificity model.....	12
Building methylation sensitivity model.....	12
The existence of secondary motif and the necessity to use composite “Min” model to predict binding energy.....	14

Introduction

HoxB13, is a homeobox protein recently found to process methylation sensitivity and two distinct motifs, i.e., TCGTAAA and CAATAAA. To quantify this, I designed following randomized dsDNA libraries covering the unmethylated, top hemimethylated, bottom hemimethylated, duplex methylated binding sites at positions 2 and 3 respectively by chemical synthesis, thus we can assay the strand-specific methyl-C contribution to the binding affinity of each site.

	Barcodes														
HoxB13-unmethylated	T	C	T	A	G	C	C	N	N	N	T	A	A	A	C
HoxB13-top hemimethylated	T	C	T	T	C	C	C	N	M	N	T	A	A	A	C
HoxB13-bottom hemimethylated	T	C	T	C	T	C	C	N	N	W	T	A	A	A	C
HoxB13-duplex methylated	T	C	T	G	A	C	C	N	M	W	T	A	A	A	C
	- 5			- 2				1			4				7

Importing and preprocessing data

```
load("../data/HoxB13.RData")
```

```
(HoxB13 <- HoxB13 %>%
  dplyr::mutate("Bound/Unbound" = Bound/Unbound,
                Energy = -log(Bound/Unbound)))
```

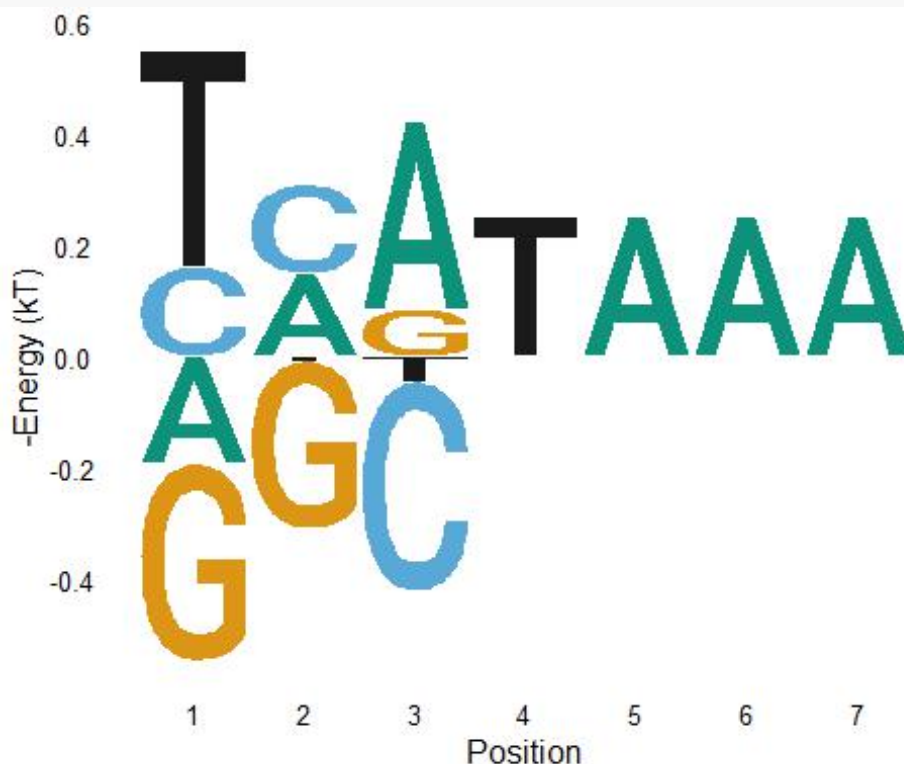
```
## # A tibble: 100 x 6
```

	Sequence	Property	Bound	Unbound	`Bound/Unbound`	Energy
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	CCATAAA	top Hemimethylated	119049	3949	30.1	-3.41
## 2	TCGTAAA	duplex Methylated	246730	16406	15.0	-2.71
## 3	TCATAAA	top Hemimethylated	113243	8850	12.8	-2.55
## 4	TCGTAAA	top Hemimethylated	157847	12818	12.3	-2.51
## 5	TCGTAAA	unmethylated	66312	12149	5.46	-1.70
## 6	ACATAAA	top Hemimethylated	42781	8102	5.28	-1.66
## 7	TCGTAAA	bottom Hemimethylated	40646	8448	4.81	-1.57
## 8	TCATAAA	unmethylated	15940	9614	1.66	-0.506
## 9	CCGTAAA	duplex Methylated	23539	15208	1.55	-0.437
## 10	CAATAAA	unmethylated	41067	27166	1.51	-0.413

```
## # ... with 90 more rows
```

Building specificity model

```
HoxB13 %>%
  dplyr::filter(Property == "unmethylated") %>%
  TFcookbook::buildEnergyModel() %>%
  TFcookbook::getEnergyMatrix() %>%
  TFcookbook::addAnchorMatrix(anchor = "TAAA", position = 4, height = 0.25) %>%
  TFcookbook::plotEnergyLogo() +
  scale_y_continuous(breaks = seq(-0.8, 0.8, 0.2))
```



Building methylation sensitivity model

It is easy to build methylation effect model by comparison of each methylated site with its unmethylated counterpart

```

HoxB13.unmethylated <- subset(HoxB13, Property == "unmethylated")
HoxB13.methylated <- subset(HoxB13, Property %in% c("top Hemimethylated", "bottom Hemimethylated", "duplex Methylated"))

(HoxB13.paired <- left_join(HoxB13.methylated, HoxB13.unmethylated, by = "Sequence")
%>%
  dplyr::mutate(Energy = Energy.x - Energy.y) %>%
  dplyr::select(Sequence,
                Property = Property.x,
                Energy))

## # A tibble: 36 x 3
##   Sequence Property      Energy
##   <chr>      <chr>      <dbl>
## 1 CCATAAA top Hemimethylated -3.08
## 2 TCGTAAA duplex Methylated -1.01
## 3 TCATAAA top Hemimethylated -2.04
## 4 TCGTAAA top Hemimethylated -0.814
## 5 ACATAAA top Hemimethylated -2.86
## 6 TCGTAAA bottom Hemimethylated 0.126
## 7 CCGTAAA duplex Methylated -1.25
## 8 CCGTAAA top Hemimethylated -0.977
## 9 TAGTAAA bottom Hemimethylated -0.632
## 10 ACGTAAA top Hemimethylated -0.902
## # ... with 26 more rows

```

Overall, we can create a compositive model to include both specificity and methylation effect

```

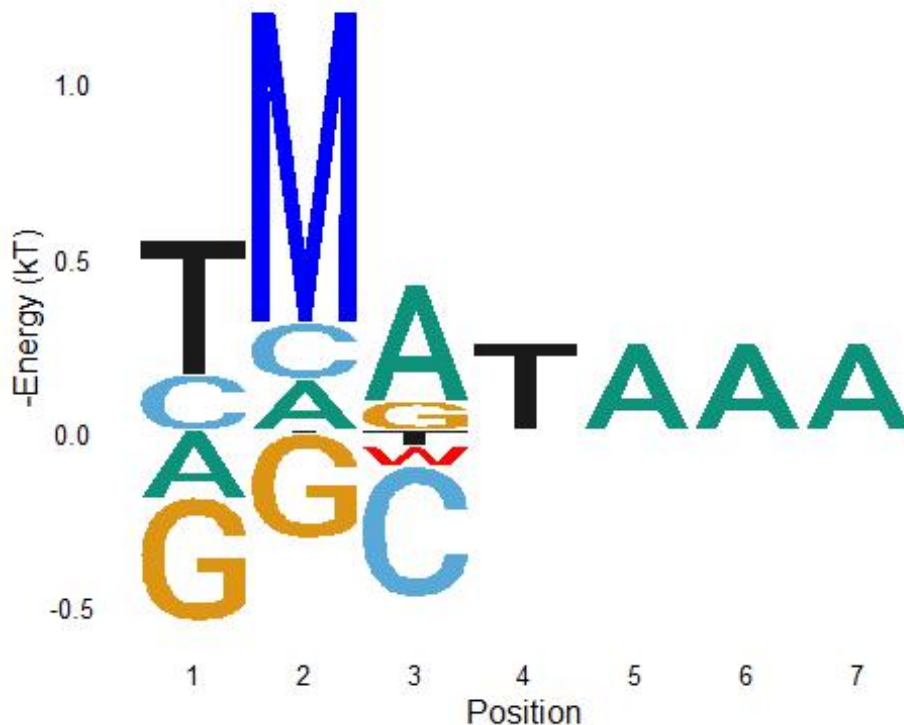
HoxB13.MethylModel <- HoxB13.paired %>%
  dplyr::mutate(Sequence = dplyr::case_when(
    Property == "top Hemimethylated" ~ stringi::stri_sub_replace(Sequence, from=2,to=2, replacement = "M"),
    Property == "bottom Hemimethylated" ~ stringi::stri_sub_replace(Sequence, from=3,to=3, replacement = "W"),
    Property == "duplex Methylated" ~ stringi::stri_sub_replace(Sequence, from=2,to=3, replacement = "MW")
  ) %>%
  TFCookbook::buildMethylationModel()

## Warning: Expected 14 pieces. Additional pieces discarded in 36 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

HoxB13 %>%
  dplyr::filter(Property == "unmethylated") %>%
  TFCookbook::buildEnergyModel() %>%
  TFCookbook::getEnergyMatrix() %>%
  TFCookbook::addAnchorMatrix(anchor = "TAAA", position = 4, height = 0.25) %>%
  TFCookbook::addMethylMatrix(MethylModel = HoxB13.MethylModel, encoding = "(3+2)L+1")
%>%
  TFCookbook::plotEnergyLogo()

## Warning: Expected 21 pieces. Additional pieces discarded in 64 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

```

The existence of secondary motif and the necessity to use composite “Min” model to predict binding energy

Since among all unmethylated sequences, TCGTAAA has the lowest binding energy and highest affinity, we can build a specificity model based on all those single variants to TCGTAAA as below. Using this model to predict the binding energy for all other sites and compare them with experimentally observed values, it is clear that small fraction of sequences, e.g., CAATAAAA and CTATAAAA, deviate significantly from the diagonal lines, meaning that they are not properly represented in the energy model made by TCGTAAA and its single variants. The most intuitive explanation is that there exists another structural conformation for the recognition of DNA sequence CAATAAAA and its related variants. We can build another model from CAAATAAAA and its single variants, called HoxB13.CAA.Model as below. However, this “CAA” model still wouldn’t explain all the observed values, particularly for sequence like TCGTAAA. Alternatively, if for each sequence we take the minimal value predicted by “TCG” and “CAA” models as the composite predicted value and compare it with the observed number, they match very well, so it is fair to say the TCG and CAA models are both required for proper understanding the specificity behavior of HoxB13 to its unmethylated sites.

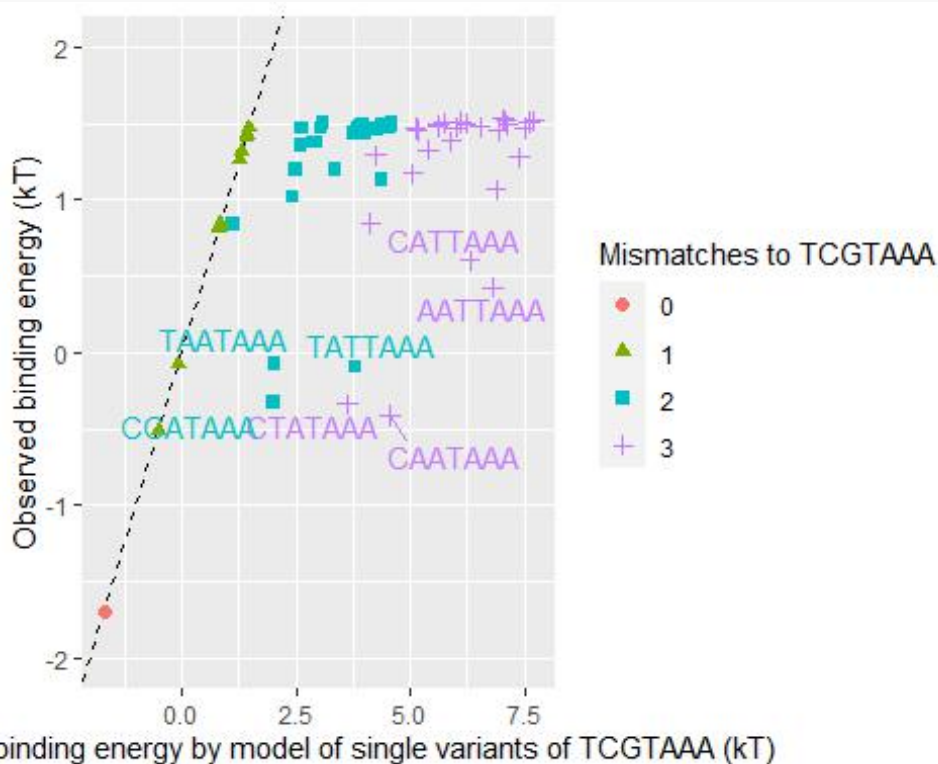
```
HoxB13.TCG.Model <- HoxB13 %>%
  dplyr::filter(Property == "unmethylated") %>%
  TFCookbook::selectVariants(reference = "TCGTAAA", maxMismatches = 1) %>%
  TFCookbook::buildEnergyModel()

HoxB13 %>%
  dplyr::filter(Property == "unmethylated") %>%
  mutate(Predicted.Energy = TFCookbook::predictEnergy(Sequence, HoxB13.TCG.Model),
         Mismatch.TCG = as.factor(TFCookbook::countMismatch(Sequence, "TCGTAAA"))) %>%
  ggplot(aes(x = Predicted.Energy, y = Energy, label = Sequence, color = Mismatch.TCG,
             shape = Mismatch.TCG)) +
  geom_abline(slop = 1, linetype = "dashed") +
```

```

geom_point(size = 2) +
geom_text_repel(data = function(x) filter(x, ((Predicted.Energy - Energy) > 2) & (E
energy < 0.8)), show.legend = FALSE) +
xlab("Predicted binding energy by model of single variants of TCGTAAA (kT)") +
ylab("Observed binding energy (kT)") + ylim(-2, 2) +
labs(color = "Mismatches to TCGTAAA", shape = "Mismatches to TCGTAAA")

```



```

HoxB13.CAA.Model <- HoxB13 %>%
  dplyr::filter(Property == "unmethylated") %>%
  TFcookbook::selectVariants(reference = "CAATAAA", maxMismatches = 1) %>%
  TFcookbook::buildEnergyModel()

```

```

HoxB13 %<>%
  dplyr::filter(Property == "unmethylated") %>%
  dplyr::mutate(Mismatch.TCG = as.factor(TFcookbook::countMismatch(Sequence, "TCGTAAA
")),
                Mismatch.CAA = as.factor(TFcookbook::countMismatch(Sequence, "CAATAAA
")),
                Predicted.Energy.TCG = TFcookbook::predictEnergy(Sequence, HoxB13.TCG.
Model),
                Predicted.Energy.CAA = TFcookbook::predictEnergy(Sequence, HoxB13.CAA.
Model)) %>%
  dplyr::mutate(Predicted.Energy.min = pmin(Predicted.Energy.TCG, Predicted.Energy.CA
A))

```

```

HoxB13 %>%
  ggplot(aes(x=Predicted.Energy.TCG, y=Energy, label=Sequence, color=Mismatch.TCG, sh
ape = Mismatch.CAA)) +
  geom_abline(slop = 1, linetype = "dashed") +
  geom_point(size = 2, show.legend = FALSE) +
  geom_text_repel(data = function(x) filter(x, ((Predicted.Energy.TCG - Energy) > 2)
& (Energy < 0.8)), show.legend = FALSE) +

```

```
xlab("Predicted binding energy\n by model of single variants of TCGTAAA (kT)") +
ylab("Observed binding energy (kT)") + ylim(-2, 2) -> plot.TCG
```

HoxB13 %>%

```
ggplot(aes(x=Predicted.Energy.CAA, y=Energy, label=Sequence, color=Mismatch.TCG, shape=Mismatch.CAA)) +
geom_abline(slop = 1, linetype = "dashed") +
geom_point(size = 2, show.legend = FALSE) +
geom_text_repel(data = function(x) filter(x, ((Predicted.Energy.CAA - Energy) > 1.5)
& (Energy < 0.8)), show.legend = FALSE) +
xlab("Predicted binding energy\n by model of single variants of CAATAAA (kT)") +
ylab("Observed binding energy (kT)") + ylim(-2, 2) -> plot.CAA
```

HoxB13 %>%

```
ggplot(aes(x = Predicted.Energy.min, y = Energy, label = substr(Sequence, 1,3), color = Mismatch.TCG, shape = Mismatch.CAA)) +
geom_abline(slop = 1, linetype = "dashed") +
geom_point(size = 2) +
geom_text_repel(data=function(x) filter(x, (abs(Predicted.Energy.min - Energy) > -0.1) & (Energy < 1)), show.legend = FALSE) +
xlab("Predicted binding energy\n by minimal value of TCGTAAA and CAATAAA models (kT)") +
ylab("Observed binding energy (kT)") + ylim(-2, 2) +
labs(color = "Mismatches to TCGTAAA", shape = "Mismatches to CAATAAA") -> plot.min
```

```
cowplot::plot_grid(plot.TCG, plot.CAA, plot.min, align = "h", ncol = 3, rel_widths = c(1,1,1.45))
```

