Primer Independence Regression

Wes Horton

June 22, 2016

Summary of Dataset and Purpose

We have approximately 170 samples per sequencing batch, and for each sample we have 260 counts, one for each of the unique combinations of V and J primers. The primers have different amplification rates, which we need to characterize. In order to do this most accurately, we need to determine if the forward (V) primer and the reverse (J) primer act independently to influence spike amplification, or if their interaction is important as well.

For this analysis, we are using 20 samples that contain only spike-ins and no DNA. They are samples 1-20 from the batch DNA160609LC, found at /home/exacloud/lustre1/CompBio/data/tcrseq/dhaarini/DNA160609LC/spike_counts/25b

Variables

1. Independent variables
    - Forward (V) primer identity - 20 total (categorical)
    - Reverse (J) primer identity - 13 total (categorical)
    - Primer combination - 260 (categorical)
2. Dependent variable
    - Spike Count

Each sample has an individual file containing the 260 counts. These need to be combined into a single data frame prior to the analysis.

```
# List spike count files, and sort by sample number
# Combine files into a data frame with 260 rows and 1 column per sample
# Melt data frame to get 1 row for each VJ + sample combination
# Take the log2 of spike counts.
DNA160609 <- read.data("/Volumes/DNA160609LC/spike_counts/25bp/spike_only_counts/")
```

Linear Regression Model

Our question is whether or not forward and reverse primers amplify independently of one another, or dependently. We can create two linear models, one for each scenario, and compare the results.

```
# Make models for with and without primer interaction
DNA160609.models <- make.models(DNA160609)
```

```
## [1] "Log2 Without Interaction R^2: 0.3753"
## [1] "Log2 With Interaction R^2: 0.8584"
```

Preliminary Results

A quick comparison of R^2 values suggests that the model which includes interaction between primers is a better fit. That model has an R^2 value of 0.8584 vs. the non-interaction model's R^2 of 0.3753. Although this suggests that it is important to include interaction effects in our model, not all of them may be giving us useful information. We can iteratively add and remove different interactions and see the effect of individual combinations of primers on our model's fit.

```
DNA160609.steps <- steps(DNA160609)
```

```
## [1] "Both: "
## Start:  AIC=-6342.22
## value ~ V + J + combos
##
##
## Step:  AIC=-6342.22
## value ~ V + combos
##
##
## Step:  AIC=-6342.22
## value ~ combos
##
##           Df Sum of Sq     RSS     AIC
## <none>                  1389.6 -6342.2
## - combos 259    8937.6 10327.2  3569.8
```

```
DNA160609.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

The step function was not working with the formula `count ~ V * J` or `count ~ V + J + V:J`. It produced an output, and picked count ~ V * J as the final model, but did not include J1-1 or V1 (and all of their combinations) in the analysis. Instead, I used the formula `count ~ V + J + combos` where combos is a new variable made by pasting the V and J primers together for a particular combination. The step function on that model concluded that count ~ combos is the best model with an R^2 of 0.8583894.

Determining important combinations

Removing unsignificant primer combinations

Only significant VJ combinations from first lm

Now that we have the step function completed, we want to step through with fewer and fewer combinations, to see if any of them are particularly important. First, we'll remove all of the VJ combinations that have p-values greater than 0.05.

```
DNA160609.first.subset <- remove.unsig.variables(DNA160609, DNA160609.steps)
DNA160609.first.subset.steps <- steps(DNA160609.first.subset)
```

```
## [1] "Both: "
## Start:  AIC=-4658.97
```

```
## value ~ V + J + combos
##
##
## Step:  AIC=-4658.97
## value ~ V + combos
##
##
## Step:  AIC=-4658.97
## value ~ combos
##
##           Df Sum of Sq    RSS     AIC
## <none>                 1020.8 -4659.0
## - combos 190      7349 8369.8  2998.3
```

```
DNA160609.first.subset.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

Again just the combos are picked for the final model, which has an R^2 of 0.8716477. So our R^2 has increased by 0.0132583 from removing 'r length(DNA160609.steps*both.summary*coefficients[,1]) - length(DNA160609.first.subset.steps*both.summary*coefficients[,1]) primer combinations of lesser importance.

Only significant VJ combinations from second lm

Let's iterate once again, using the same p-value threshold and the new p-values produced by the subsetted model.

```
DNA160609.second.subset <- remove.unsig.variables(DNA160609, DNA160609.first.subset.steps)
DNA160609.second.subset.steps <- steps(DNA160609.second.subset)
```

```
## [1] "Both: "
## Start:  AIC=-4517.66
## value ~ V + J + combos
##
##
## Step:  AIC=-4517.66
## value ~ V + combos
##
##
## Step:  AIC=-4517.66
## value ~ combos
##
##           Df Sum of Sq    RSS     AIC
## <none>                  987.4 -4517.7
## - combos 184    6659.2 7646.6  2687.9
```

```
DNA160609.second.subset.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

The R^2 value actually went down to 0.8641083 this time. So our R^2 has increased by 0.0057189 from removing 75 primer combinations of lesser importance.

Only significant VJ combinations from third lm

Let's subset a final time and run the step function, to see what we get.

```
DNA160609.third.subset <- remove.unsig.variables(DNA160609, DNA160609.second.subset.steps)
DNA160609.third.subset.steps <- steps(DNA160609.third.subset)
```

```
## [1] "Both: "
## Start:  AIC=-4028.8
## value ~ V + J + combos
##
##
## Step:  AIC=-4028.8
## value ~ V + combos
##
##
## Step:  AIC=-4028.8
## value ~ combos
##
##           Df Sum of Sq    RSS     AIC
## <none>                  880.8 -4028.8
## - combos 164    6151.2 7032.0  2498.6
```

```
DNA160609.third.subset.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

The R^2 is relatively unchanged at 0.8681909. From this, we can see that the first subset increased R^2 the most, although even that increase was modest. The combinations removed during the first step are the least important for determining spike counts. There are 68 of them:

```
DNA160609.first.subset$unsig.coeff
```

```
##                        Estimate Std. Error     t value   Pr(>|t|)
## combosV1-J1-4       0.279024247  0.1677175  1.66365645 0.09624453
## combosV1-J1-6       0.174224793  0.1677175  1.03879933 0.29894891
## combosV1-J2-5      -0.195929257  0.1677175 -1.16821020 0.24277836
## combosV12-1-2-J1-5 -0.223843786  0.1677175 -1.33464802 0.18205308
## combosV12-1-2-J2-1  0.134836696  0.1677175  0.80395142 0.42146372
## combosV12-1-2-J2-4  0.017057535  0.1677175  0.10170399 0.91899577
## combosV12-1-2-J2-5  0.165504343  0.1677175  0.98680445 0.32378684
## combosV13-1-J1-1    0.162031844  0.1677175  0.96609999 0.33404144
## combosV13-1-J1-2    0.068430997  0.1677175  0.40801354 0.68328142
## combosV13-1-J1-3    0.073845399  0.1677175  0.44029641 0.65974170
## combosV13-1-J1-4    0.042019578  0.1677175  0.25053788 0.80218184
## combosV13-1-J1-6    0.059721951  0.1677175  0.35608665 0.72179088
## combosV13-1-J2-2    0.139249412  0.1677175  0.83026183 0.40643088
## combosV13-1-J2-7   -0.272124358  0.1677175 -1.62251650 0.10475661
## combosV13-2-J1-1   -0.321096230  0.1677175 -1.91450679 0.05561314
## combosV13-2-J1-2    0.046066614  0.1677175  0.27466796 0.78358284
## combosV13-2-J1-4    0.071193356  0.1677175  0.42448385 0.67123147
## combosV13-2-J1-6    0.063257318  0.1677175  0.37716595 0.70606648
```

```
## combosV13-2-J2-1    0.050790302  0.1677175  0.30283251 0.76203025
## combosV13-2-J2-2    0.315941850  0.1677175  1.88377428 0.05965420
## combosV13-2-J2-3    0.079955501  0.1677175  0.47672734 0.63357738
## combosV13-2-J2-4   -0.186907552  0.1677175 -1.11441912 0.26515364
## combosV13-2-J2-7   -0.216824115  0.1677175 -1.29279388 0.19614282
## combosV13-3-J2-1   -0.247144561  0.1677175 -1.47357675 0.14065930
## combosV13-3-J2-3   -0.055243282  0.1677175 -0.32938300 0.74188018
## combosV13-3-J2-5    0.060038085  0.1677175  0.35797157 0.72037988
## combosV14-J1-1      0.132381861  0.1677175  0.78931469 0.42996598
## combosV14-J1-6     -0.257998376  0.1677175 -1.53829163 0.12404137
## combosV14-J1-7      0.328296991  0.1677175  1.95744067 0.05035191
## combosV15-J2-7     -0.007851951  0.1677175 -0.04681654 0.96266134
## combosV16-J1-1     -0.021474266  0.1677175 -0.12803834 0.89812384
## combosV16-J1-2      0.243772566  0.1677175  1.45347155 0.14615640
## combosV16-J1-6     -0.211937403  0.1677175 -1.26365731 0.20641268
## combosV16-J1-7      0.198890377  0.1677175  1.18586561 0.23573241
## combosV16-J2-1     -0.166449911  0.1677175 -0.99244231 0.32103041
## combosV16-J2-4      0.251345267  0.1677175  1.49862308 0.13403534
## combosV16-J2-5     -0.052507432  0.1677175 -0.31307074 0.75424013
## combosV17-J1-2      0.159631647  0.1677175  0.95178904 0.34125055
## combosV17-J1-5     -0.222350617  0.1677175 -1.32574514 0.18498543
## combosV17-J1-7     -0.190702533  0.1677175 -1.13704635 0.25557406
## combosV17-J2-3     -0.071548279  0.1677175 -0.42660004 0.66968926
## combosV19-J1-6     -0.097631975  0.1677175 -0.58212169 0.56051128
## combosV19-J1-7      0.253228447  0.1677175  1.50985136 0.13114531
## combosV19-J2-7     -0.279299939  0.1677175 -1.66530024 0.09591628
## combosV2-J1-2       0.165510280  0.1677175  0.98683985 0.32376949
## combosV2-J2-1      -0.143332899  0.1677175 -0.85460925 0.39280895
## combosV2-J2-3      -0.254633663  0.1677175 -1.51822984 0.12902046
## combosV23-J1-1     -0.295255635  0.1677175 -1.76043462 0.07839602
## combosV23-J1-7     -0.270339003  0.1677175 -1.61187148 0.10705379
## combosV23-J2-2     -0.215187686  0.1677175 -1.28303682 0.19953937
## combosV23-J2-4      0.070265826  0.1677175  0.41895354 0.67526830
## combosV23-J2-5      0.068020502  0.1677175  0.40556600 0.68507908
## combosV24-J2-1      0.123851433  0.1677175  0.73845280 0.46027442
## combosV24-J2-2      0.153906904  0.1677175  0.91765578 0.35884390
## combosV24-J2-4      0.052711417  0.1677175  0.31428698 0.75331636
## combosV24-J2-7     -0.315507511  0.1677175 -1.88118457 0.06000558
## combosV29-J1-4      0.003528740  0.1677175  0.02103979 0.98321477
## combosV29-J2-4     -0.176819718  0.1677175 -1.05427134 0.29181027
## combosV29-J2-5     -0.017077668  0.1677175 -0.10182403 0.91890048
## combosV3-J1-1      -0.161512128  0.1677175 -0.96300123 0.33559404
## combosV3-J1-7      -0.112695442  0.1677175 -0.67193623 0.50165572
## combosV3-J2-2       0.067342928  0.1677175  0.40152602 0.68805025
## combosV30-J2-5     -0.047365504  0.1677175 -0.28241247 0.77763909
## combosV4-J1-2       0.222840909  0.1677175  1.32866846 0.18401875
## combosV4-J1-7      -0.326097921  0.1677175 -1.94432891 0.05191249
## combosV4-J2-2      -0.287960564  0.1677175 -1.71693843 0.08605311
## combosV5-J1-2      -0.256328647  0.1677175 -1.52833602 0.12649315
## combosV5-J1-7      -0.242862604  0.1677175 -1.44804598 0.14766765
```

Removing significant combinations

Instead of removing unsignificant combinations and looking for an increase in R^2 to identify unimportant primer combinations, we can go in the reverse direction. In the following sections, I will be removing significant primer combinations (based on varying thresholds) and looking for a decrease in R^2. A significant decrease in R^2 will imply that the significant primer combinations that were excluded from the model are important in determining spike counts.

Remove VJ combinations with p < 0.001

First, we'll use a modest cut-off of 0.001. This means we will remove all primer combinations from our dataset whose p-values from our original model were less than 0.001. The remaining data that we will enter into the linear model are our least significant values.

```
DNA160609.first.subset.b <- remove.sig.variables(DNA160609, DNA160609.steps, 0.001)
DNA160609.first.subset.b.steps <- steps(DNA160609.first.subset.b)
```

```
## [1] "Both: "
## Start:  AIC=-2446.68
## value ~ V + J + combos
##
##
## Step:  AIC=-2446.68
## value ~ V + combos
##
##
## Step:  AIC=-2446.68
## value ~ combos
##
##          Df Sum of Sq    RSS     AIC
## <none>                532.49 -2446.7
## - combos 99    150.14 682.63 -2147.9
```

```
DNA160609.first.subset.b.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

Removing these 160 primer combinations significantly dropped the R^2 value from 0.8583894 to 0.1792931, sugggesting that all of these primer combinations are important to our model. We removed 160 combinations here, which is over half of the total combinations, and not very informative. In addition, each removal of a primer combination is approximately associated with a 0.0042444 decrease in R^2.

Remove VJ combinations with p < 10^{-5}

Let's try again, subsetting from the original data frame, but using a more stringent p-value cut-off of 0.00001. This time, the data that we will model will be all primer combinations whose p-values in the original model were greater than 10^{-5}.

```
DNA160609.second.subset.b <- remove.sig.variables(DNA160609, DNA160609.steps, 0.00001)
DNA160609.second.subset.b.steps <- steps(DNA160609.second.subset.b)
```

```
## [1] "Both: "
## Start:  AIC=-3255.34
## value ~ V + J + combos
##
##
## Step:  AIC=-3255.34
## value ~ V + combos
##
##
## Step:  AIC=-3255.34
## value ~ combos
##
##           Df Sum of Sq     RSS     AIC
## <none>                  707.88 -3255.3
## - combos 132    416.28 1124.15 -2289.1
```

```
DNA160609.second.subset.b.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

For this iteration, we removed 127 primer combinations and now have an R^2 of: 0.3374096. Additionally, each removal of a primer combination is approximately associated with a 0.0041022 decrease in R^2.

Remove VJ combinations with $p < 10^{-7}$

Let's subset once again, with an even more stringent p-value cut-off of 10^{-7}. Once again, the data that we are modelling will be all primer combinations whose p-values from the original model were greater than 10^{-7}.

```
DNA160609.third.subset.b <- remove.sig.variables(DNA160609, DNA160609.steps, 0.0000001)
DNA160609.third.subset.b.steps <- steps(DNA160609.third.subset.b)
```

```
## [1] "Both: "
## Start:  AIC=-3644
## value ~ V + J + combos
##
##
## Step:  AIC=-3644
## value ~ V + combos
##
##
## Step:  AIC=-3644
## value ~ combos
##
##           Df Sum of Sq     RSS   AIC
## <none>                  793.82 -3644
## - combos 148    612.87 1406.69 -2235
```

```
DNA160609.third.subset.b.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

This subset gives us an R^2 of 0.4061799. For this subset, we removed 111 primer combinations. Once again, each removal of a primer combination is approximately associated with a 0.004074 decrease in R^2.

Remove VJ combinations with $p < 10^{-16}$

There are quite a few 2 < 10e-16 combinations, what does our model look like if we use everything but those? In other words, if we remove our most significant primer combinations only, do we have a significant drop in R^2?

```
DNA160609.fourth.subset.b <- remove.sig.variables(DNA160609, DNA160609.steps, 0.000000000000001)
DNA160609.fourth.subset.b.steps <- steps(DNA160609.fourth.subset.b)
```

```
## [1] "Both: "
## Start:  AIC=-4476.25
## value ~ V + J + combos
##
##
## Step:  AIC=-4476.25
## value ~ V + combos
##
##
## Step:  AIC=-4476.25
## value ~ combos
##
##           Df Sum of Sq     RSS     AIC
## <none>                  974.77 -4476.2
## - combos 182      1330 2304.79 -1690.7
```

```
DNA160609.fourth.subset.b.steps$both.summary$call
```

```
## lm(formula = value ~ combos, data = batch$log2)
```

This subset gives us an R^2 of 0.5549299. For this subset, we removed 77 primer combinations, and each removal is approximately associated with a 0.003941 decrease in R^2.

Summary of removing significant combinations

Our first subset resulted in the most drastic drop in R^2 as well as the greatest per-combo-removed drop. I don't know how to determine if the drops/increases in R^2 are due to the identities of the primer combinations, or the fact that we're removing significant portions of the dataset.

Testing each extermely significant combinations individually

There are 77 combinations with p-values less than 10^{-16}. Let's see what happens if we iterate over all of these VJ combinations, removing them one at a time. The data sets will be the full data set, minus a single primer combination.

```
# Remove one-by-one and record adj.r.square values produced.
compare.r2s <- iterate.sig.combos(DNA160609, DNA160609.steps)
# Look at the range of values produced
summary(as.numeric(compare.r2s$R2))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8536  0.8574  0.8582  0.8578  0.8586  0.8588
```

```
# Look at 5 smalles adj.r.sq values:
sorted.r2 <- compare.r2s[order(as.numeric(compare.r2s$R2)),]
lowest.5 <- sorted.r2[1:5,]
lowest.5
```

```
##    Primer.Combo                R2
## 44     V26-J2-3 0.853597026524842
## 18     V16-J2-2 0.854183107908046
## 36     V20-J2-2 0.854277167381199
## 72      V5-J2-4 0.854593998586895
## 73      V5-J2-5 0.855083017982819
```

From the range of R^2 values, we see that no single primer combination provides an undue boost to the model. We can look at the V and J identities of these primer combinations and see if anythign stands out.

```
# Look at the individual V's and J's that make up the 77 most important combinations
compare.r2s$V <- gsub("-J.*", "", compare.r2s$Primer.Combo)
compare.r2s$J <- gsub("V.*J", "J", compare.r2s$Primer.Combo)
vcounts <- count(compare.r2s, "V")
jcounts <- count(compare.r2s, "J")
vcounts[order(vcounts$freq, decreasing = T),]
```

```
##          V freq
## 16      V4    8
## 12     V26    7
## 5      V15    6
## 8      V19    6
## 10     V20    6
## 14      V3    6
## 15     V30    6
## 17      V5    6
## 2  V12-1-2    5
## 6      V16    3
## 7      V17    3
## 9       V2    3
## 1       V1    2
## 4      V14    2
## 11     V23    2
## 3    V13-2    1
## 13     V29    1
```

```
jcounts[order(jcounts$freq, decreasing = T),]
```

```
##       J freq
## 1  J1-3   11
## 2  J1-4    9
## 3  J1-5    8
## 11 J2-7    8
## 6  J2-1    7
## 8  J2-3    7
## 9  J2-4    7
## 7  J2-2    6
```

```
## 10 J2-5    6
## 4  J1-6    3
## 5  J1-7    1
```

Looks like V4 and J1-3 are involved in the most significant combinations.

Conclusions

From this report, it apprears that primer combination is important for determining spike counts. Additionally, there does not appear to be any particular primer combination that increases the model fit more than any of the other combinations. Based of this analsysis, future normalization methods will have to take primer interactions into consideration and create 260 scaling factors, not just 33.