DNA160609_spike_only_contamination

Wes Horton

July 14, 2016

Overview

In the control batch DNA160609LC, samples 1-20 contain only spike-ins and primers (no DNA). During our pipeline, we run a spike removal tool that searches for a 9-bp barcode within fastq reads and removes those reads. After running this program on the spike-only samples, we still have reads in these samples.

We need to find out what proportion of total reads are not spikes, as well as what the source of these reads are. One likely cause is from the p14 DNA that contaminated the batch. They could also be spikes that were not removed in the spike removal step.

Set up

We need the PEAR'ed fastq files for samples 1-20. These will act as the baseline counts for each sample. The despiked fastq files produced by the spike removal tool will be our comparison counts. Lastly, we'll need the exported alignment files in order to determine p14 contamination.

Analysis

Using these files, we'll determine

1. The proportion of reads that aren't spikes
2. The proportion of non-spiked reads that are from p14
3. The proportion of reads that aren't spikes, but also are not p14 (unaccounted reads)

```
##  [1] 0.8684770 0.7461521 0.6332091 0.5951745 0.5924477 0.6092125 0.5999344
##  [8] 0.4702209 0.5948057 0.5691646 0.5466388 0.5635463 0.6388492 0.7837993
## [15] 0.6802170 0.7016472 0.7637548 0.6541527 0.6115455 0.5104863
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4702  0.5866  0.6104  0.6367  0.6856  0.8685
```

```
##  [1] 0.016775709 0.011079105 0.018254838 0.015337423 0.011680878
##  [6] 0.014916468 0.011623852 0.011407712 0.012655024 0.016294606
## [11] 0.009543806 0.017550018 0.020370748 0.025094103 0.015885624
## [16] 0.019884669 0.016708438 0.022603978 0.016747613
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.009544 0.012170 0.016290 0.016020 0.017900 0.025090
```

Only a small percentage of the total reads are not the spike-ins (less than 1%). Of these non-spike reads, a small amount are accounted for by the p14 contamination, with the rest of unkown origin.

```
##  [1] 0.8683313 0.7460694 0.6330936 0.5950832 0.5923785 0.6091216 0.5998647
##  [8] 0.4701673 0.5947305 0.5690718 0.5465866 0.5634474 0.6387190 0.7836026
## [15] 0.6801089 0.7015077 0.6540434 0.6114073 0.5104008
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.4702  0.5807  0.6091  0.6299  0.6671  0.8683
```

Removing p14 contamination doesn't significantly change the proportion of contaminated reads. Are there any primer combinations that stand out as being overrepresented?

```
##     sample total.count Best V hit Best J hit   N proportion
##  1:      1        5531      V13-3       J2-4 246   4.447659
##  2:      1        5531        V20       J2-1 231   4.176460
##  3:      1        5531      V13-1       J1-6  93   1.681432
##  4:      1        5531        V24       J1-6  84   1.518713
##  5:      1        5531      V13-2       J2-1  83   1.500633
##  6:      2        8311      V13-3       J2-4 321   3.862351
##  7:      2        8311        V20       J2-1 256   3.080255
##  8:      2        8311      V12-1       J1-2 136   1.636386
##  9:      2        8311      V13-1       J2-4 122   1.467934
## 10:      2        8311      V13-3       J2-5 121   1.455902
## 11:      3        4968      V13-1       J2-1  76   1.529791
## 12:      3        4968      V13-1       J2-5  75   1.509662
## 13:      3        4968        V20       J2-1  75   1.509662
## 14:      3        4968      V13-1       J2-4  75   1.509662
## 15:      3        4968         V1       J1-2  68   1.368760
## 16:      4        5702         V1       J1-2 110   1.929148
## 17:      4        5702      V13-3       J2-4 106   1.858997
## 18:      4        5702      V12-1       J1-2  98   1.718695
## 19:      4        5702      V13-1       J2-1  91   1.595931
## 20:      4        5702      V13-1       J2-5  85   1.490705
## 21:      5        7844        V20       J2-1 127   1.619072
## 22:      5        7844      V12-1       J1-2 120   1.529832
## 23:      5        7844         V1       J1-2 111   1.415094
## 24:      5        7844      V13-1       J2-1 103   1.313106
## 25:      5        7844      V13-2       J2-2  99   1.262111
## 26:      6        6250      V12-1       J1-2 101   1.616000
## 27:      6        6250        V20       J2-1  94   1.504000
## 28:      6        6250      V13-1       J2-3  92   1.472000
## 29:      6        6250      V13-1       J1-6  92   1.472000
## 30:      6        6250      V13-3       J2-4  89   1.424000
## 31:      7        7837      V13-3       J2-4 170   2.169197
## 32:      7        7837      V12-1       J1-2 129   1.646038
## 33:      7        7837      V13-1       J2-4 120   1.531198
## 34:      7        7837         V1       J1-2 116   1.480158
## 35:      7        7837        V20       J2-1 114   1.454638
## 36:      8        7887      V12-1       J1-2 134   1.698998
## 37:      8        7887         V1       J1-2 123   1.559528
## 38:      8        7887      V13-3       J2-4 122   1.546849
## 39:      8        7887      V13-1       J2-3 115   1.458096
## 40:      8        7887      V13-1       J2-4 113   1.432737
## 41:      9        6939      V13-3       J2-4 216   3.112840
## 42:      9        6939         V1       J1-2 137   1.974348
## 43:      9        6939      V13-1       J2-4 127   1.830235
## 44:      9        6939      V13-1       J2-3 114   1.642888
## 45:      9        6939      V13-1       J2-1 114   1.642888
## 46:     10        5617      V13-3       J2-4 183   3.257967
## 47:     10        5617      V13-1       J2-1  84   1.495460
## 48:     10        5617      V13-1       J2-4  82   1.459854
## 49:     10        5617      V13-1       J1-6  80   1.424248
## 50:     10        5617        V24       J1-6  77   1.370839
## 51:     11        9475      V13-3       J2-4 248   2.617414
## 52:     11        9475      V12-1       J1-2 163   1.720317
## 53:     11        9475      V13-1       J2-5 150   1.583113
## 54:     11        9475         V1       J1-2 139   1.467018
## 55:     11        9475      V13-1       J2-1 134   1.414248
## 56:     12        5095      V13-3       J2-4 144   2.826300
## 57:     12        5095      V12-1       J1-2  88   1.727184
## 58:     12        5095      V13-1       J2-4  80   1.570167
## 59:     12        5095      V13-1       J2-1  76   1.491658
## 60:     12        5095      V13-2       J1-6  74   1.452404
## 61:     13        4434        V20       J2-1 169   3.811457
## 62:     13        4434      V13-3       J2-4  82   1.849346
## 63:     13        4434      V13-1       J2-5  74   1.668922
## 64:     13        4434         V1       J1-2  72   1.623816
## 65:     13        4434      V13-1       J2-4  69   1.556157
## 66:     14        3520      V13-3       J2-4  93   2.642045
## 67:     14        3520        V20       J2-1  77   2.187500
## 68:     14        3520      V12-1       J1-2  67   1.903409
## 69:     14        3520      V13-1       J2-1  58   1.647727
## 70:     14        3520      V13-1       J2-5  49   1.392045
## 71:     15        5723      V13-3       J2-4 123   2.149222
## 72:     15        5723      V12-1       J1-2  97   1.694915
## 73:     15        5723        V20       J2-1  96   1.677442
## 74:     15        5723      V13-1       J2-1  82   1.432815
## 75:     15        5723      V13-1       J2-3  81   1.415342
## 76:     16        4628         V1       J1-2  82   1.771824
## 77:     16        4628        V20       J2-1  79   1.707001
## 78:     16        4628      V13-1       J2-4  77   1.663786
## 79:     16        4628      V12-1       J1-2  63   1.361279
## 80:     16        4628      V13-3       J2-4  63   1.361279
## 81:     18        5479      V13-3       J2-4  84   1.533126
## 82:     18        5479      V12-1       J1-2  84   1.533126
## 83:     18        5479        V20       J2-1  80   1.460120
## 84:     18        5479      V13-1       J2-4  79   1.441869
## 85:     18        5479         V1       J1-2  77   1.405366
## 86:     19        3968      V13-3       J2-4  73   1.839718
## 87:     19        3968      V13-1       J2-1  69   1.738911
## 88:     19        3968         V1       J1-2  69   1.738911
## 89:     19        3968        V20       J2-1  54   1.360887
## 90:     19        3968      V13-1       J1-6  53   1.335685
## 91:     20        5307      V13-3       J2-4 129   2.430752
## 92:     20        5307      V13-1       J2-4  96   1.808932
## 93:     20        5307         V1       J1-2  87   1.639344
## 94:     20        5307      V13-1       J2-1  85   1.601658
## 95:     20        5307      V12-1       J1-2  79   1.488600
##     sample total.count Best V hit Best J hit   N proportion
```

None of the primer combinations have particularly high proportions. One thing of note, however, is that V13-1, V13-2, and V13-3 seem to appear often. We can group by V's instead of V/J and see if any particular V's are messing things up.

```
##     sample total.count Best V hit    N proportion
##  1:      1        5531      V13-3  591  10.685229
##  2:      1        5531      V13-1  533   9.636594
##  3:      1        5531        V24  482   8.714518
##  4:      1        5531      V13-2  438   7.919002
##  5:      1        5531         V1  375   6.779967
##  6:      2        8311      V13-1  871  10.480087
##  7:      2        8311      V13-3  831   9.998797
##  8:      2        8311        V24  725   8.723379
##  9:      2        8311      V13-2  652   7.845025
## 10:      2        8311         V1  610   7.339670
## 11:      3        4968      V13-1  543  10.929952
## 12:      3        4968        V24  443   8.917069
## 13:      3        4968      V13-2  432   8.695652
## 14:      3        4968         V1  408   8.212560
## 15:      3        4968      V13-3  351   7.065217
## 16:      4        5702      V13-1  640  11.224132
## 17:      4        5702        V24  508   8.909155
## 18:      4        5702         V1  499   8.751315
## 19:      4        5702      V13-2  470   8.242722
## 20:      4        5702      V13-3  427   7.488600
## 21:      5        7844      V13-1  788  10.045895
## 22:      5        7844        V24  673   8.579806
## 23:      5        7844      V13-2  641   8.171851
## 24:      5        7844         V1  635   8.095360
## 25:      5        7844      V12-1  534   6.807751
## 26:      6        6250      V13-1  672  10.752000
## 27:      6        6250        V24  590   9.440000
## 28:      6        6250         V1  516   8.256000
## 29:      6        6250      V13-2  497   7.952000
## 30:      6        6250      V13-3  409   6.544000
## 31:      7        7837      V13-1  836  10.667347
## 32:      7        7837         V1  664   8.472630
## 33:      7        7837        V24  638   8.140870
## 34:      7        7837      V13-3  621   7.923950
## 35:      7        7837      V13-2  608   7.758071
## 36:      8        7887      V13-1  877  11.119564
## 37:      8        7887      V13-2  661   8.380880
## 38:      8        7887        V24  659   8.355522
## 39:      8        7887         V1  643   8.152656
## 40:      8        7887      V13-3  537   6.808672
## 41:      9        6939      V13-1  887  12.782822
## 42:      9        6939         V1  682   9.828506
## 43:      9        6939      V13-3  674   9.713215
## 44:      9        6939      V13-2  595   8.574723
## 45:      9        6939        V24  557   8.027093
## 46:     10        5617      V13-1  629  11.198148
## 47:     10        5617      V13-3  529   9.417839
## 48:     10        5617        V24  496   8.830336
## 49:     10        5617      V13-2  479   8.527684
## 50:     10        5617         V1  458   8.153819
## 51:     11        9475      V13-1 1061  11.197889
## 52:     11        9475      V13-3  823   8.686016
## 53:     11        9475      V13-2  812   8.569921
## 54:     11        9475          V1  785   8.284960
## 55:     11        9475         V24  773   8.158311
## 56:     12        5095       V13-1  597  11.717370
## 57:     12        5095       V13-3  477   9.362120
## 58:     12        5095         V24  439   8.616290
## 59:     12        5095       V13-2  424   8.321884
## 60:     12        5095          V1  415   8.145240
## 61:     13        4434       V13-1  511  11.524583
## 62:     13        4434          V1  412   9.291836
## 63:     13        4434       V13-2  361   8.141633
## 64:     13        4434       V13-3  354   7.983762
## 65:     13        4434         V24  345   7.780785
## 66:     14        3520       V13-1  403  11.448864
## 67:     14        3520         V24  308   8.750000
## 68:     14        3520       V13-2  307   8.721591
## 69:     14        3520       V13-3  305   8.664773
## 70:     14        3520          V1  287   8.153409
## 71:     15        5723       V13-1  636  11.113053
## 72:     15        5723       V13-2  493   8.614363
## 73:     15        5723         V24  443   7.740695
## 74:     15        5723       V13-3  441   7.705749
## 75:     15        5723          V1  412   7.199021
## 76:     16        4628       V13-1  483  10.436474
## 77:     16        4628         V24  403   8.707865
## 78:     16        4628          V1  377   8.146067
## 79:     16        4628       V13-2  369   7.973207
## 80:     16        4628       V12-1  311   6.719965
## 81:     18        5479       V13-1  596  10.877897
## 82:     18        5479          V1  474   8.651214
## 83:     18        5479         V24  469   8.559956
## 84:     18        5479       V13-2  451   8.231429
## 85:     18        5479       V13-3  368   6.716554
## 86:     19        3968       V13-1  428  10.786290
## 87:     19        3968          V1  363   9.148185
## 88:     19        3968       V13-2  337   8.492944
## 89:     19        3968         V24  313   7.888105
## 90:     19        3968       V13-3  293   7.384073
## 91:     20        5307       V13-1  627  11.814585
## 92:     20        5307       V13-3  450   8.479367
## 93:     20        5307          V1  447   8.422838
## 94:     20        5307         V24  435   8.196721
## 95:     20        5307       V13-2  432   8.140192
##      sample total.count Best V hit    N proportion
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   41.70   42.95   43.82   44.20   44.98   48.93
```

```
##    Best V hit  N
## 1:      V13-3 17
## 2:      V13-1 19
## 3:        V24 19
## 4:      V13-2 19
## 5:         V1 19
## 6:      V12-1  2
```

The same 5 V's are almost always the top 5, and they generally make up about 43.8204509 percent of all of the reads. Is this the same in the spike count files?

```
##       V4  N
## 1: V13-1 19
## 2:    V1 19
## 3:   V29 19
## 4:   V24 19
## 5:   V23 16
## 6: V13-2  3
```

The unique V's that are in the top 5 in at least one sample are not the same between the left-over reads and the spiked reads, although 3 of them are the same. In fact, V13-1, V1, and V24 are in the top 5 in all samples. Let's try and see if they're primer-dimers or some chimeric read caused by mis-amplification. Process:

1. For each entry in an alignment file, extract the V and J hits
2. Use V and J to extract appropriate 34-bp synthetic template from spike file
3. Divide synthetic template into 6 9-bp strings ([1:9], [5:14], [10:19], [15:24], [20:29], [25:34])
4. Search the fastq read of the alignment entry for these strings
5. Observe distribution of hits

```
# Each entry in the align file has a V and J as well as a sequence.
# Can take the V and J identities and search for them in the spike file.
# Then take the spike sequence from the spike file.
# Split into a few substrings (first try strings of length 9, every 5)
# Use vcountPattern to check if they're there.
spikes <- read.table("~/Desktop/OHSU/tcr_spike/text_barcodesvj.txt", header = T, sep = ' ',
                     stringsAsFactors = F)
spikes.v122 <- spikes[spikes$V == "V12-1-2-",]
spikes.v122$V <- gsub("V12-1-2-", "V12-2", spikes.v122$V)
spikes$V <- gsub("V12-1-2-", "V12-1", spikes$V)
spikes$V <- gsub("-$", "", spikes$V)

test <- list()
for (i in 1:length(align.files)){
  curr.align <- suppressWarnings(fread(paste(align.dir, align.files[i], sep = ''), na.strings = c('', '
                     showProgress = F))
  curr.align$`Best V hit` <- gsub("TRB|\\*00", '', curr.align$`Best V hit`)
  curr.align$`Best J hit` <- gsub("TRB|\\*00", '', curr.align$`Best J hit`)
  index <- gsub(".*_S|_align.*", '', align.files[i])
  align.query.results <- NULL
  for (j in 1:length(curr.align$`Read(s) sequence`)){
    V <- curr.align$`Best V hit`[j]
    J <- curr.align$`Best J hit`[j]
    if (V %in% spikes$V && J %in% spikes$J){
      fastq.read <- curr.align$`Read(s) sequence`[j]
      query <- spikes[spikes$V == V & spikes$J == J, "SPIKE"]
      query <- unlist(strsplit(query, split = ''))
      sub.query <- c(paste(query[1:9], collapse = ''), paste(query[5:14], collapse = ''),
                    paste(query[10:19], collapse = ''), paste(query[15:24], collapse = ''),
                    paste(query[20:29], collapse = ''), paste(query[25:34], collapse = ''))
      query.results <- vector(mode = "numeric", length = 6)
      for (k in 1:length(sub.query)){
        query.results[k] <- vcountPattern(sub.query[k], fastq.read)
      } # for k
      align.query.results <- rbind(align.query.results, query.results)
    } # if
  } # for j
  test[[i]] <- align.query.results
} # for i

# Now we have a list containing 19 data frames with nrow = number of alignments and ncol = 6 (one for e
# For each list, take the row sum and add it as a column
for (i in 1:length(test)){
  test[[i]] <- data.table(test[[i]])
  test[[i]]$sum <- apply(test[[i]], 1, sum)
}

results <- matrix(nrow = length(test), ncol = 9)
for (i in 1:length(test)){
  pass.4 <- length(test[[i]][test[[i]]$sum >= 4,`sum`])
  pass.3 <- length(test[[i]][test[[i]]$sum >= 3,`sum`])
  pass.2 <- length(test[[i]][test[[i]]$sum >= 2,`sum`])
  pass.1 <- length(test[[i]][test[[i]]$sum >= 1,`sum`])
  total <- length(test[[i]][,`sum`])
  proportion.4 <- round(pass.4 / total * 100, digits = 2)
  proportion.3 <- round(pass.3 / total * 100, digits = 2)
  proportion.2 <- round(pass.2 / total * 100, digits = 2)
  proportion.1 <- round(pass.1 / total * 100, digits = 2)
  new.row <- c(total, pass.4, proportion.4, pass.3, proportion.3, pass.2, proportion.2, pass.1, proport
  results[i,] <- new.row
}
colnames(results) <- c("total.reads", "Reads.4.hits", "Proportion.4", "Reads.3.hits",
                       "Proportion.3", "Reads.2.hits", "Proportion.2", "Reads.1.hit", "Proportion.1")
rownames(results) <- c(1:16, 18:20)
results
```

```
##    total.reads Reads.4.hits Proportion.4 Reads.3.hits Proportion.3
## 1         5507         1154        20.96         1668        30.29
## 2         8278         1974        23.85         2782        33.61
## 3         4955         1452        29.30         2033        41.03
## 4         5680         1622        28.56         2333        41.07
## 5         7831         2227        28.44         3222        41.14
## 6         6235         1674        26.85         2441        39.15
## 7         7809         2189        28.03         3064        39.24
## 8         7867         2509        31.89         3494        44.41
## 9         6908         1880        27.21         2634        38.13
## 10        5604         1606        28.66         2254        40.22
## 11        9451         2838        30.03         3949        41.78
## 12        5080         1505        29.63         2101        41.36
## 13        4412         1131        25.63         1618        36.67
## 14        3503          870        24.84         1227        35.03
## 15        5715         1609        28.15         2261        39.56
## 16        4623         1278        27.64         1818        39.33
## 18        5465         1540        28.18         2237        40.93
## 19        3956         1212        30.64         1673        42.29
## 20        5296         1607        30.34         2254         42.56
##    Reads.2.hits Proportion.2 Reads.1.hit Proportion.1
## 1          2239        40.66        2688        48.81
## 2          3700        44.70        4442        53.66
## 3          2644        53.36        3108        62.72
## 4          3100        54.58        3707        65.26
## 5          4201        53.65        4968        63.44
## 6          3188        51.13        3779        60.61
## 7          3999        51.21        4753        60.87
## 8          4516        57.40        5307        67.46
## 9          3466        50.17        4129        59.77
## 10         2929        52.27        3445        61.47
## 11         5075        53.70        6010        63.59
## 12         2671        52.58        3109        61.20
## 13         2113        47.89        2549        57.77
## 14         1569        44.79        1885        53.81
## 15         2967        51.92        3507        61.36
## 16         2417        52.28        2892        62.56
## 18         2860        52.33        3439        62.93
## 19         2176        55.01        2554        64.56
## 20         2959        55.87        3488        65.86
```

Using this method, it looks like 50-60 percent of the reads are some version of spikes. One suggestion from this is to incorporate the secondary barcode from the spike file during our spike removal step. A spike sequence is is 34-bp long, the first 9 are a universal spike barcode, the last 9 are a different universal barcode, and the remaining 16 are unique identifiers for each individual spike.

Currently, we identify spikes for removal using the first 9-bp sequence only. We could potentially use both the first and the second 9-bp sequences to remove spikes.