

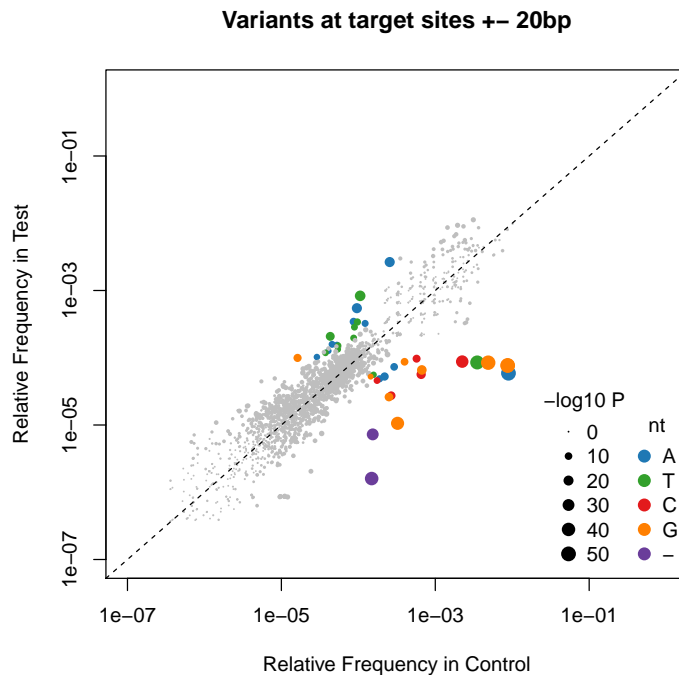
1 DeepSNV

Going to use DeepSNV to analyze the amplicon deep sequencing data. I'm not sure where the primers begin and end, so I'm looking at the variant sites ± 20 bp. It is necessary to include sites other than the variants themselves to estimate the overdispersion parameter of the model.

```
> library(deepSNV)
> marco_excel_file <- read.delim('C05C08-Met.txt')
> regions <- with(marco_excel_file, data.frame(chr=CHR, start=POS-20, stop=POS+20))
> #primary <- deepSNV(test='../data/marco_amplicon/C05C08-primary.sort.RG.q40.bam',
> #                  control='../data/marco_amplicon/C05C08-normal.sort.RG.q40.bam',
> #                  regions=regions, alternative="two.sided")
> #save(primary, file='primary_twosided.Rdata')
>
> load('primary_twosided.Rdata')
> primary.bb <- estimateDispersion(primary, alternative="two.sided")
```

Note: The initial object used a binomial model. Will be changed to beta-binomial.
Estimated dispersion factor 34.3303149518501

```
> plot(primary.bb)
> title("Variants at target sites  $\pm 20$ bp")
```

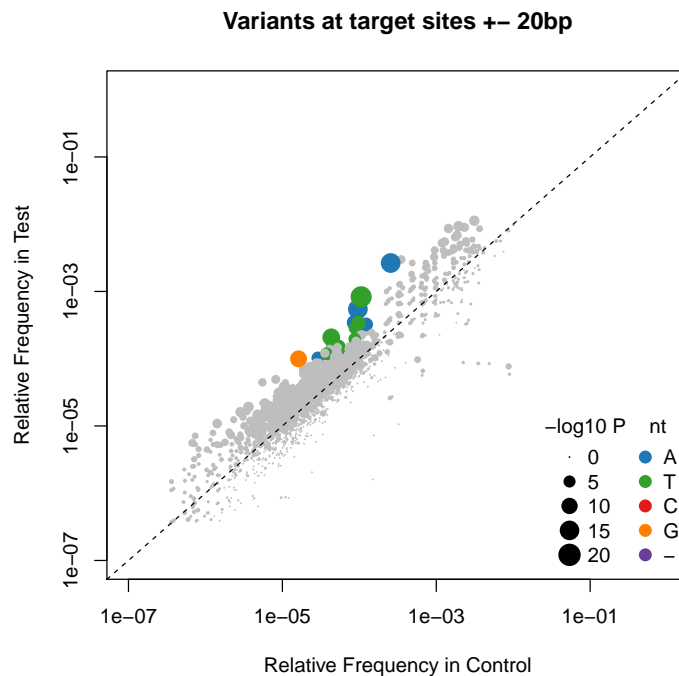


Looking at the two-sided test shows an extremely high false positive rate. If we're going to use these though we need to do one-sided tests: when reporting a negative result, we need to be as sensitive as possible.

```
> #primary <- deepSNV(test='../data/marco_amplicon/C05C08-primary.sort.RG.q40.bam',
> #                      control='../data/marco_amplicon/C05C08-normal.sort.RG.q40.bam',
> #                      regions=regions, alternative="greater")
> #save(primary, file='primary_onesided.Rdata')
> #
> load('primary_onesided.Rdata')
> primary.bb <- estimateDispersion(primary, alternative="greater")
```

Note: The initial object used a binomial model. Will be changed to beta-binomial.
Estimated dispersion factor 24.5487267838615

```
> plot(primary.bb)
> title("Variants at target sites +- 20bp")
```



Many variants detected, probably all false positives. We are only interested in those at targeted sites.

```
> calls <- summary(primary.bb, sig.level=.1, adjust.method="BH")
```

With this many false positives, a negative result is meaningful. Is a positive result meaningful?

```
> # Chance of a false positive landing on a targeted site by chance
> # (assuming all calls are false positives)
> nrow(calls) / (nrow(marco_excel_file) * 41)

[1] 0.04645761
```

Yes, it is. There are lots of false positives, but if you view the statistical test as a black box and do a permutation test, a positive at a targeted site is $p < .05$. A positive at a targeted site *for the targeted mutation* is even more significant. So, what do we have?

```
> is.targeted <- with(calls, sprintf("%d:%d", chr, pos)) %in%
+               with(marco_excel_file, sprintf("%d:%d", CHR, POS))
> calls[is.targeted,]

      chr      pos ref var      p.val      freq.var sigma2.freq.var n.tst.fw
29 chr2 11716651  G   C 0.0657390130 2.422012e-05 3.677649e-11      21
4 chr22 46930524  C   A 0.0004395565 7.330804e-05 8.971379e-11     101
      cov.tst.fw n.tst.bw cov.tst.bw n.ctrl.fw cov.ctrl.fw n.ctrl.bw cov.ctrl.bw
29      580262      20      590461      9      780539      8      793393
4       686168      37      670185     29      982739     26     951471
      raw.p.val
29 6.895220e-04
4 1.531556e-06
```

Two significant sites. Are they the same variants as were found in the metastasis?

```
> is.targeted.2 <- with(marco_excel_file, sprintf("%d:%d", CHR, POS)) %in%
+               with(calls, sprintf("%d:%d", chr, pos))
> marco_excel_file[is.targeted.2,]
```

	CHR	POS	REF	VAR	MET_REF_Count	MET_VAR_Count	MET_TOTAL_Count
9	chr2	11716651	G	C	79	27	106
10	chr22	46930524	C	T	113	92	205

The chr22 mutation site is called as C>A by DeepSNV, but we were looking for C>T. However the chr2 site is called as G>C, which is what we were looking for.

In order to update the supplementary table, we need the p values for each site. The adjusted p values from the above summary are not appropriate, since they are adjusted for over 800 comparisons, but we are only reporting about 20. I'm not sure whether adjusting for even these 20 is appropriate. On the one hand, p value adjustment procedures tend to be conservative, and thus not really appropriate for reporting a negative result, since you can get as few significant hits as you like by using a more conservative procedure. On the other hand, with over 20 tests, on average you should get at least one positive, even if the null hypothesis is true in every case. The probability of at least one false positive at a .05 significance level:

```
> 1-dbinom(0, nrow(marco_excel_file), .05)

[1] 0.6594384

> 1-dbinom(0, nrow(marco_excel_file), .05) - dbinom(1, nrow(marco_excel_file), .05)

[1] 0.2830282
```

More likely than not. And there's as likely to be 2 as 0. I think the best thing to do is just report the p values, noting that a false positive or two is expected.

There's two ways to go about producing the list of p values. One is to report the p values associated with each site. The other is to report the p value associated with specifically the variant you are looking for. Loading the variant specific p values. Looking at them, you can see that they correspond to the variant calls made with the summary function:

```
> is.targeted.3 <- rep(c(rep(FALSE, 20), TRUE, rep(FALSE, 20)), nrow(marco_excel_file))
> targeted.p.vals <- primary.bb@p.val[is.targeted.3,]
> targeted.p.vals[is.targeted.2,]
```

	A	T	C	G -
[1,]	8.973063e-02	0.3431475	0.000689522	NA 1
[2,]	1.531556e-06	0.3411477	NA	0.05817626 1

We have the G>C mutation represented with a .000689 p value for C, and the C>A mutation represented with a .0000015 p value for A. The expected C>T mutation, on the other hand, has a p value of .34. Retrieving all these p values:

```
> targeted.var.p.vals <- mapply(``,
+                               lapply(1:nrow(targeted.p.vals), function(i) targeted.p.vals[i,]),
+                               as.character(marco_excel_file$VAR))
> targeted.var.p.vals
```

	A	G	G	C	T	C
0.846573590	0.216653460	1.000000000	0.596573590	0.124077842	0.345142699	
	T	A	C	T	T	A
0.846573590	1.000000000	0.000689522	0.341147711	0.838476316	0.512890212	
	C	G	A	C	T	C
1.000000000	1.000000000	0.499686185	1.000000000	1.000000000	0.846573590	
	G	T	T			
0.846573590	0.135600977	1.000000000				

It is interesting to note just *how* low the significant p value is. This is not something that can be passed off as a false positive expected under uniformly distributed p values. We can check this with a statistical test, using the fact that the minimum of a uniform distribution has a Beta(1,n) distribution:

```
> pbeta(min(targeted.var.p.vals), 1, length(targeted.var.p.vals))
```

```
[1] 0.01438055
```

Thus the chr22 variant p value is lower than what would be expected from uniformly distributed p values.

Also, I bet that result stands up to adjustment for multiple comparisons. Looking at the strictest one that's reasonable, the Bonferroni correction:

```
> p.adjust(targeted.var.p.vals, method="bonferroni")
```

	A	G	G	C	T	C	T
1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
	A	C	T	T	A	C	G
1.00000000	0.01447996	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
	A	C	T	C	G	T	T
1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000

Anyway, getting the site-specific ones rather than variant-specific ones would be hard and it's not really what we're looking for, we're looking for the p value *of the targeted variant*, so I'll just output this.

```
> writeLines(as.character(targeted.var.p.vals), "targeted_variant_p_values.txt")
```