In silico R4IDs simulation

Aim: establish likely performance.

Methods

A python script was used to simulate nanopore R9 sequencing runs from published **Cammelinae** genomes (~/Documents/all_work/programming/oddjects-sandbox/R4IDs/In-silico-genome-skimming-by-args.py). This used a mutation rate of 1/20 e.g. 5%. Uniformly-selected sites for mutation were assigned a substitution (equal base frequencies), point deletion, or point insertion (homopolymer created by inserting **n**th base as **n+1**th)

These read sets (intensities 10, 100, 1000 and 10000 reads) were used to simulate both R4IDs set-up and ID resequencing runs as for the empirical sci-fest data, and analysed in the same way.

Input reference genomes

Species	File	Mbp	# contigs
A. thaliana	()	120	7
Capsella rubella	file:////Users/joeparker/Downloads/ANNY01.1.fsa_nt.gz	129	7,067
A. halleri subsp. gemmifera	file:////Users/joeparker/Downloads/FJVB01.1.fsa_nt.gz	196	2,239
A. lyrata	()	208	3645
Capsella bursa-pastoris	file:////Users/joeparker/Downloads/MPGU01.1.fsa_nt.gz	268	8,186

Species	File	Mbp	# contigs
Camelina sativa	file:////Users/joeparker/Downloads/JFZQ01.1.fsa_nt.gz file:////Users/joeparker/Downloads/JFZQ01.2.fsa_nt.gz	641	37,780

Nanopore sequencing simulation

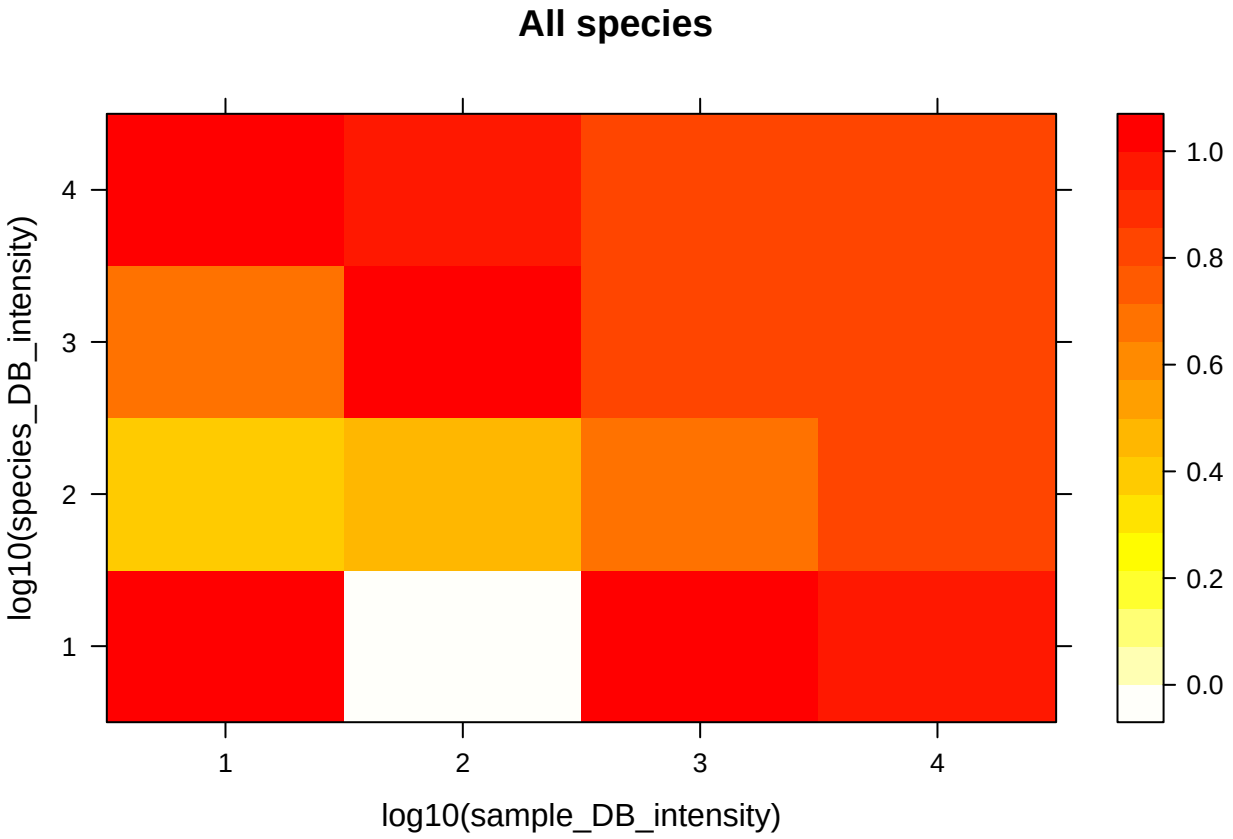
Analysis

```
library(lattice)
in_silico = read.table('~/Documents/all_work/programming/oddjects-sandbox/R4IDs/in-silico.out.parsed.td
```

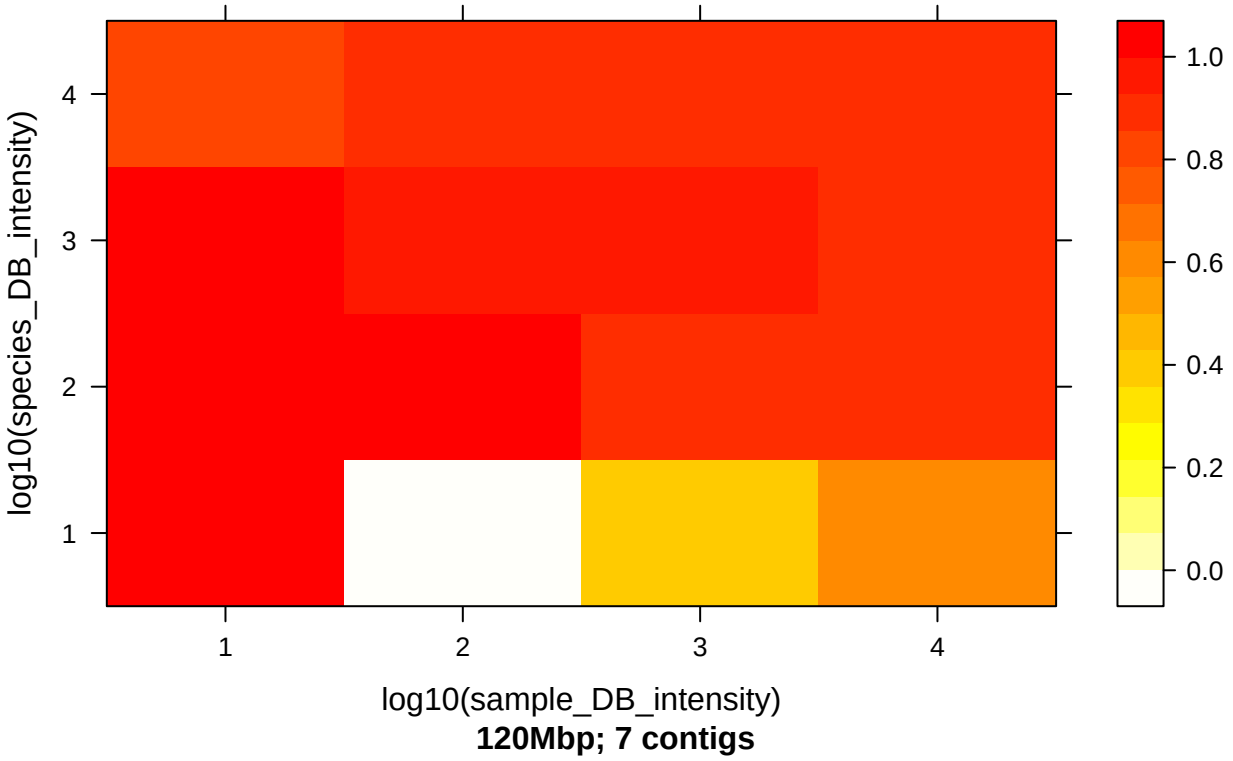
How do stats' performance vary by species? List species in order.

First: *rate TP: total*

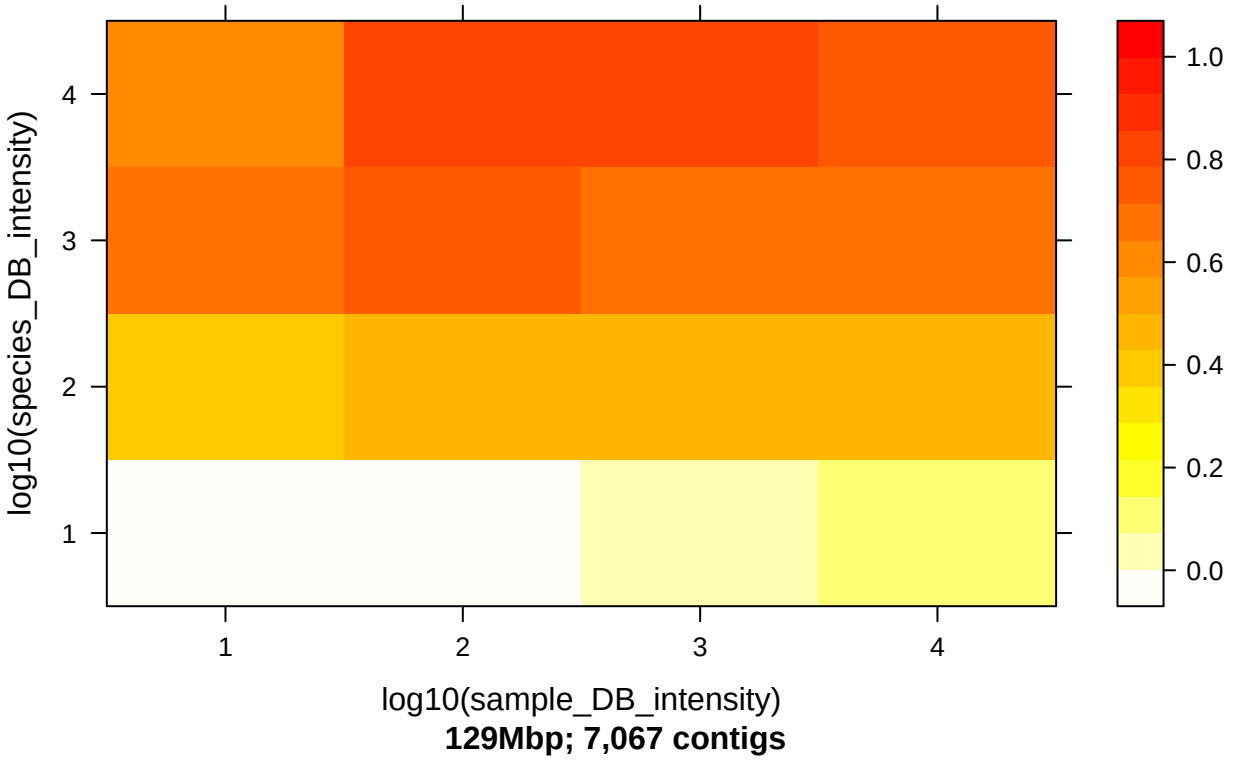
```
levelplot((one_way_TP + two_way_TP) / total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensit
```



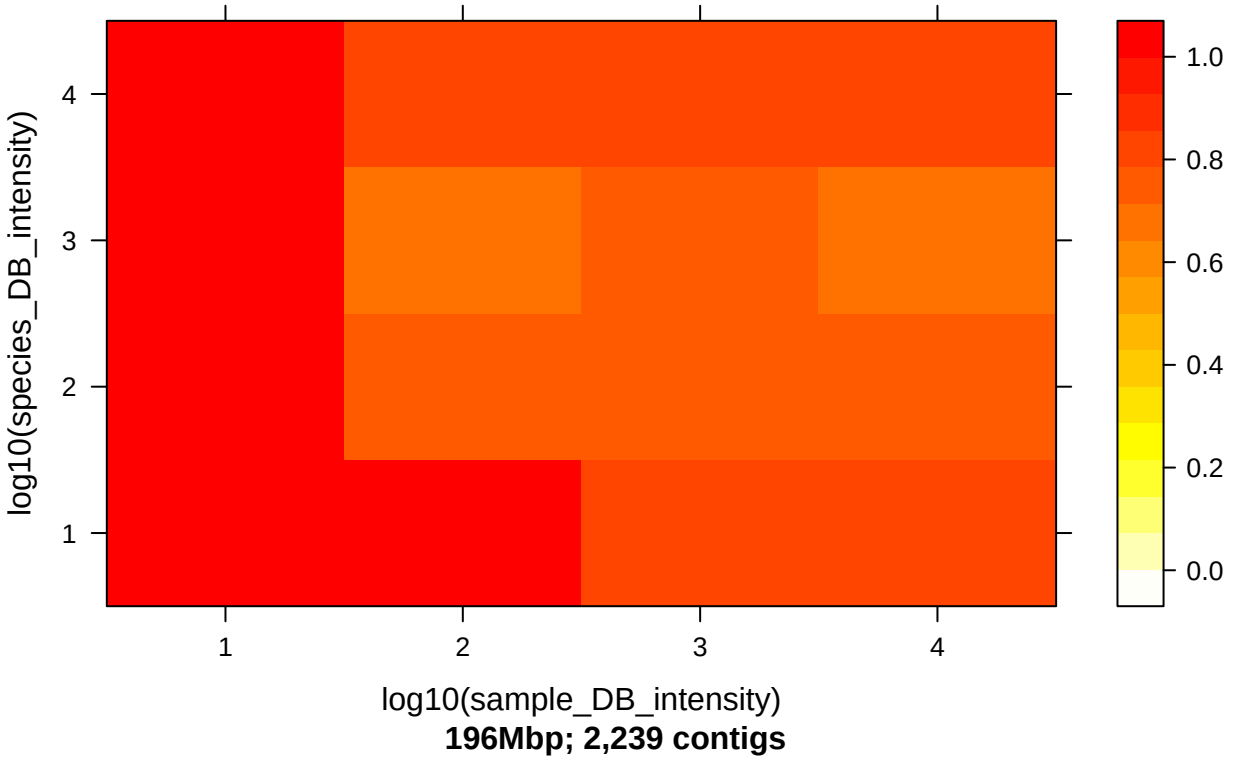

```
levelplot((one_way_TP + two_way_TP) / total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity
```



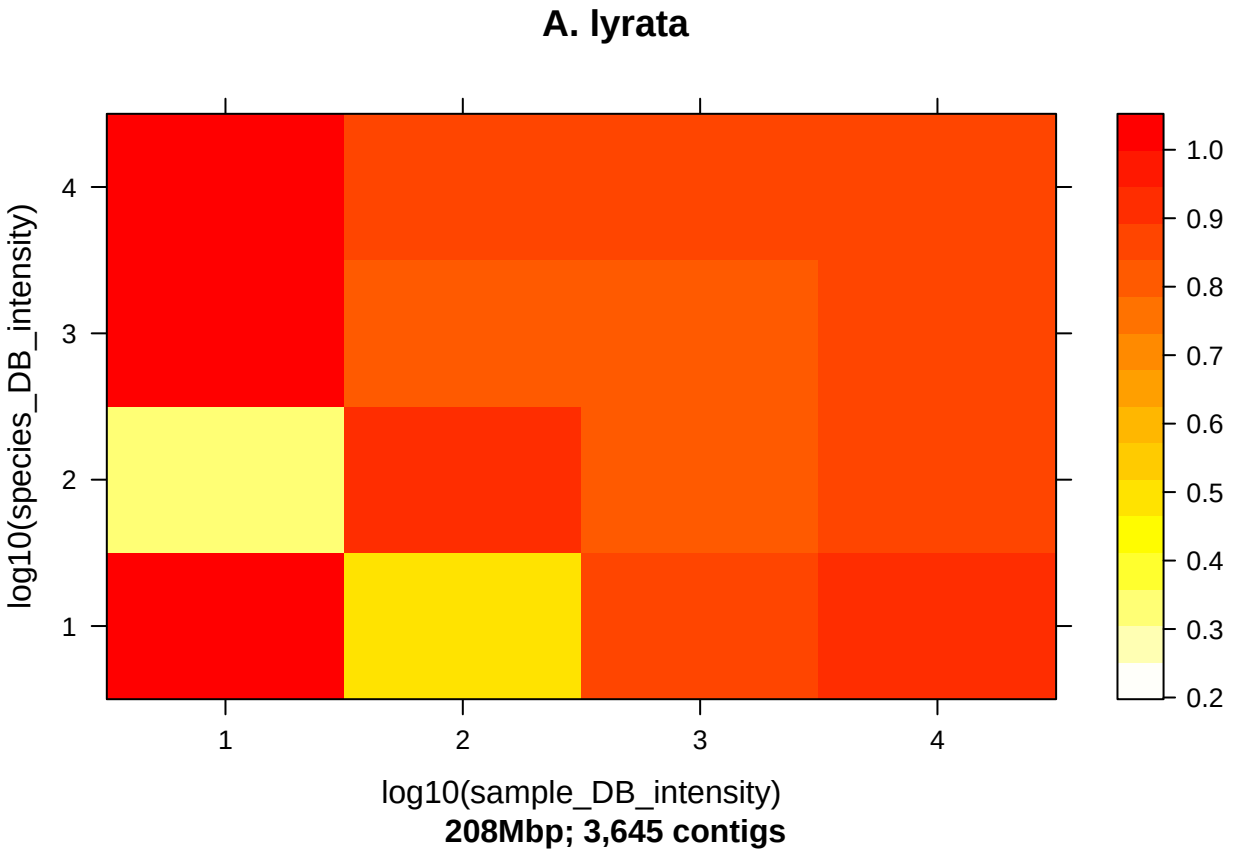

```
levelplot((one_way_TP + two_way_TP) / total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity
```



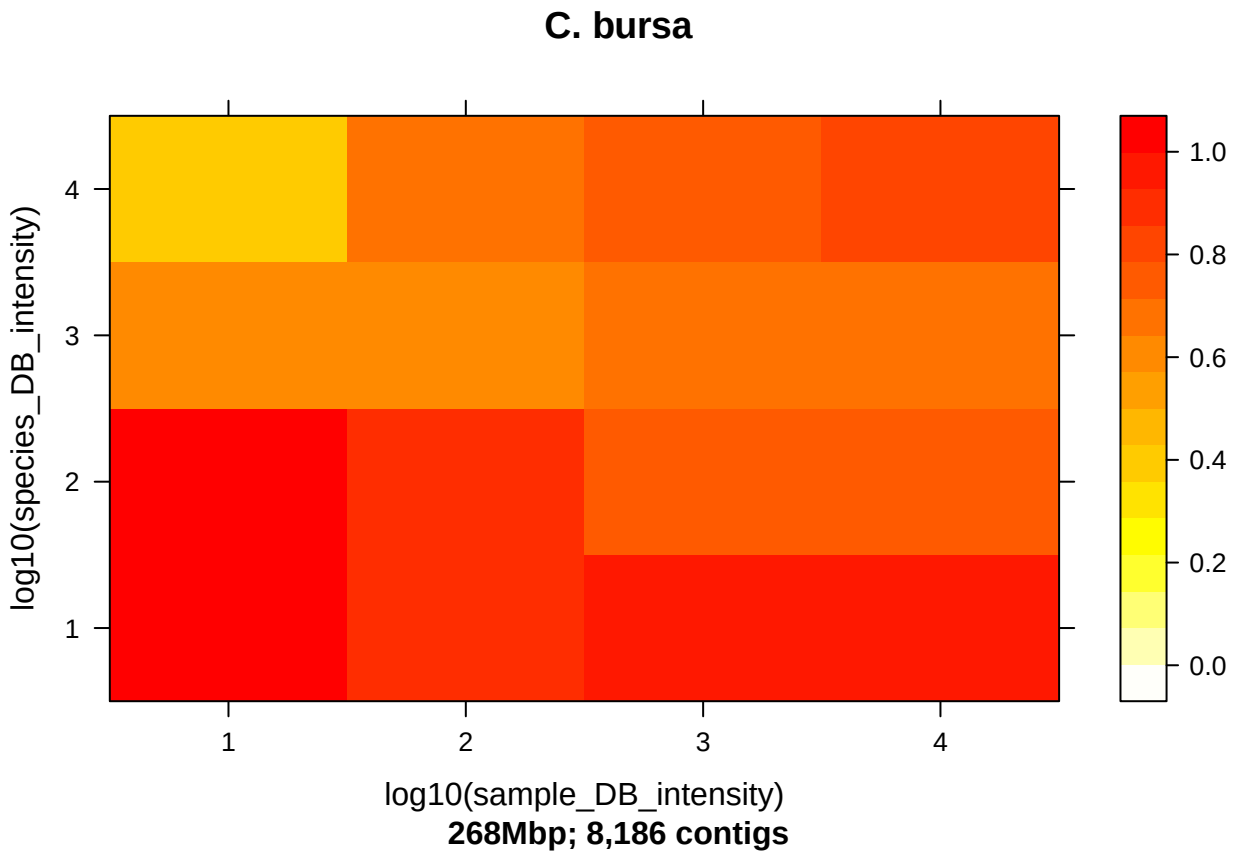

```
levelplot((one_way_TP + two_way_TP) / total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity
```



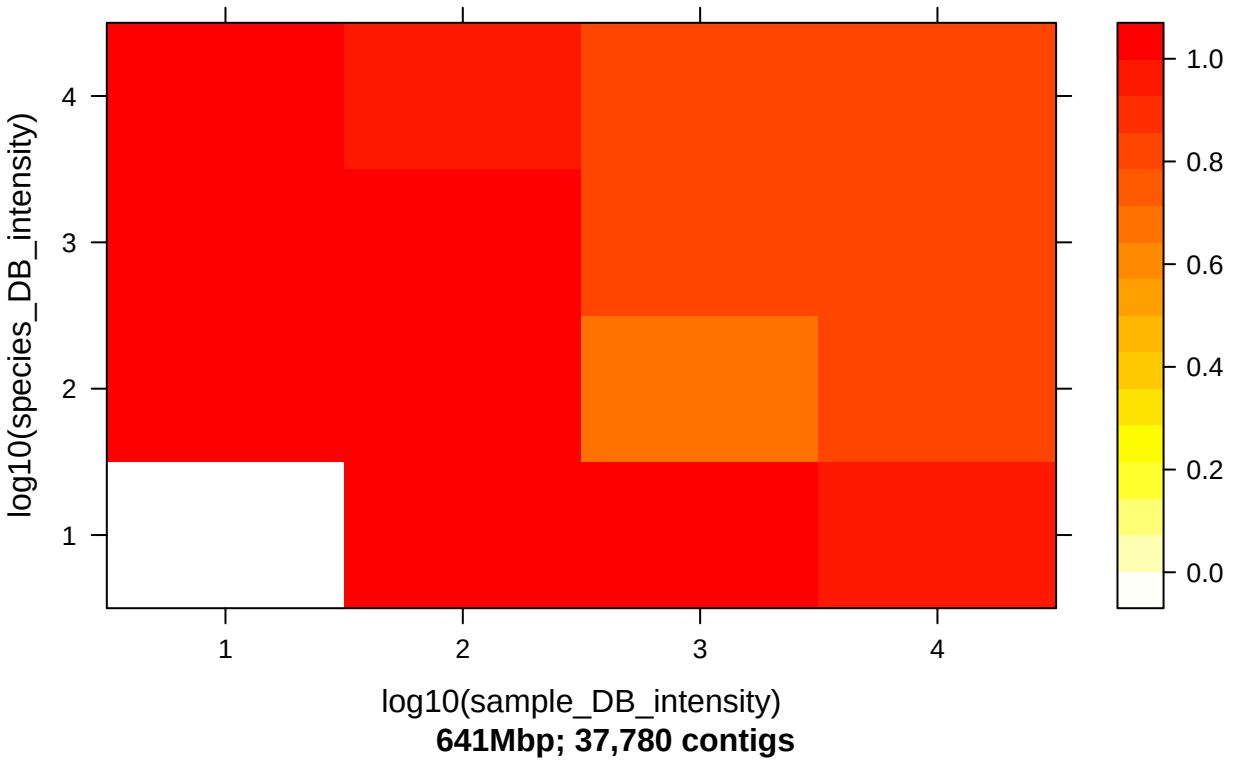

```
levelplot((one_way_TP + two_way_TP) / total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity
```




```
levelplot((one_way_TP + two_way_TP) / total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity
```

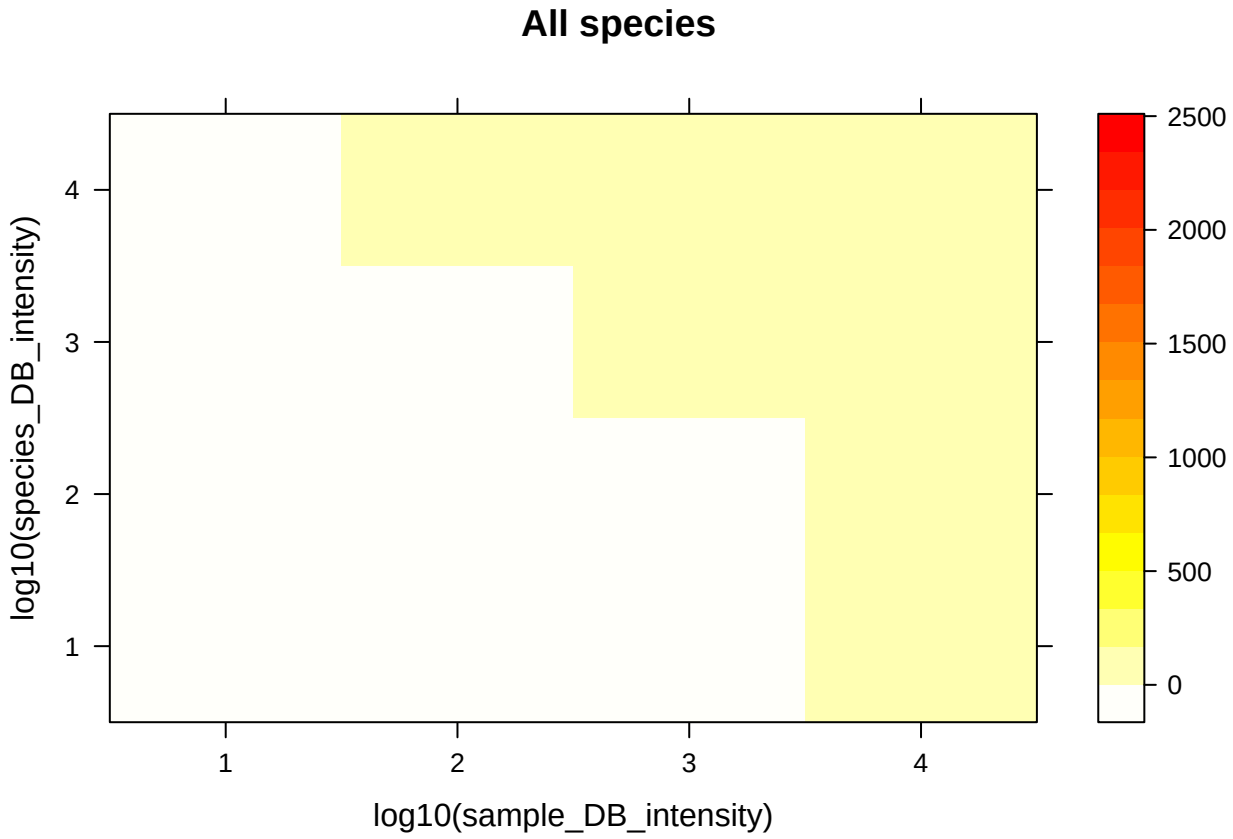



```
levelplot((one_way_TP + two_way_TP) / total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity
```

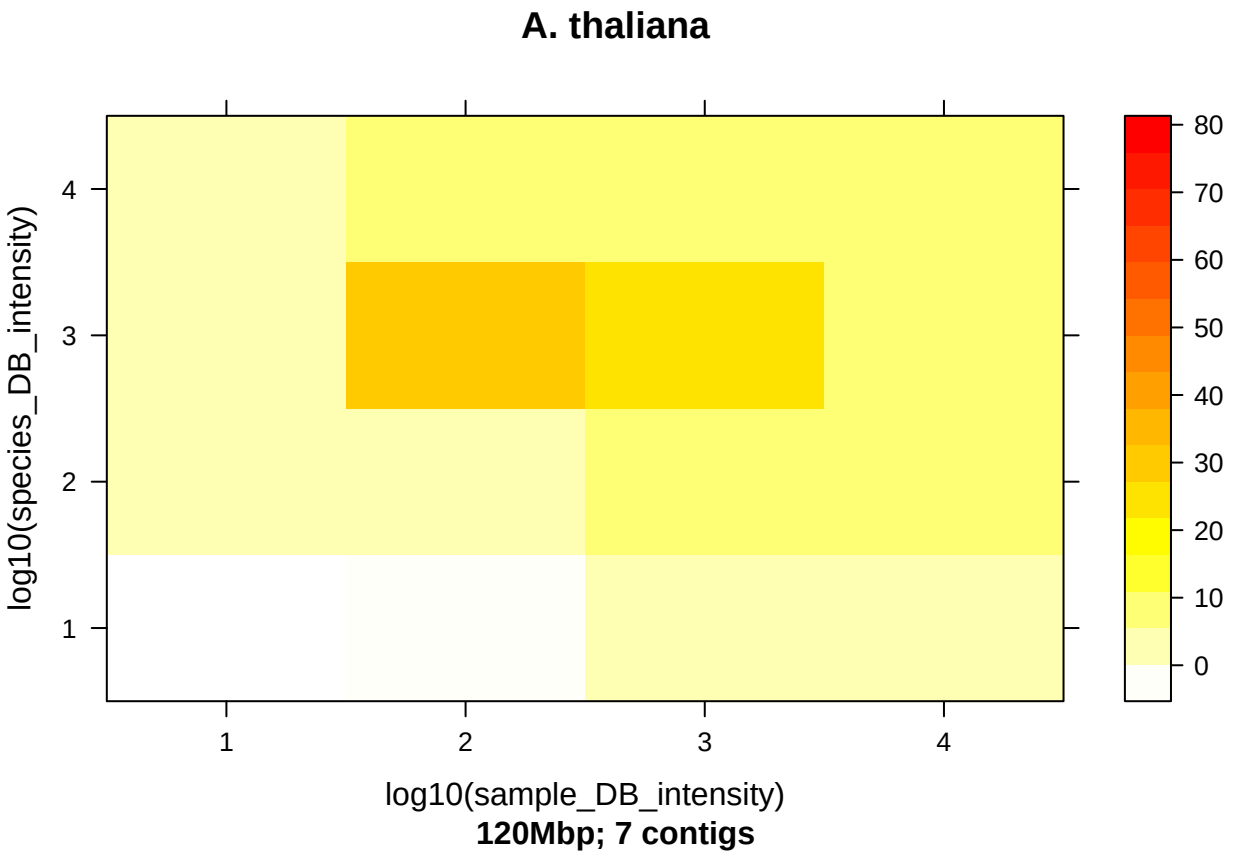


Secondly *rate TP:FP*:

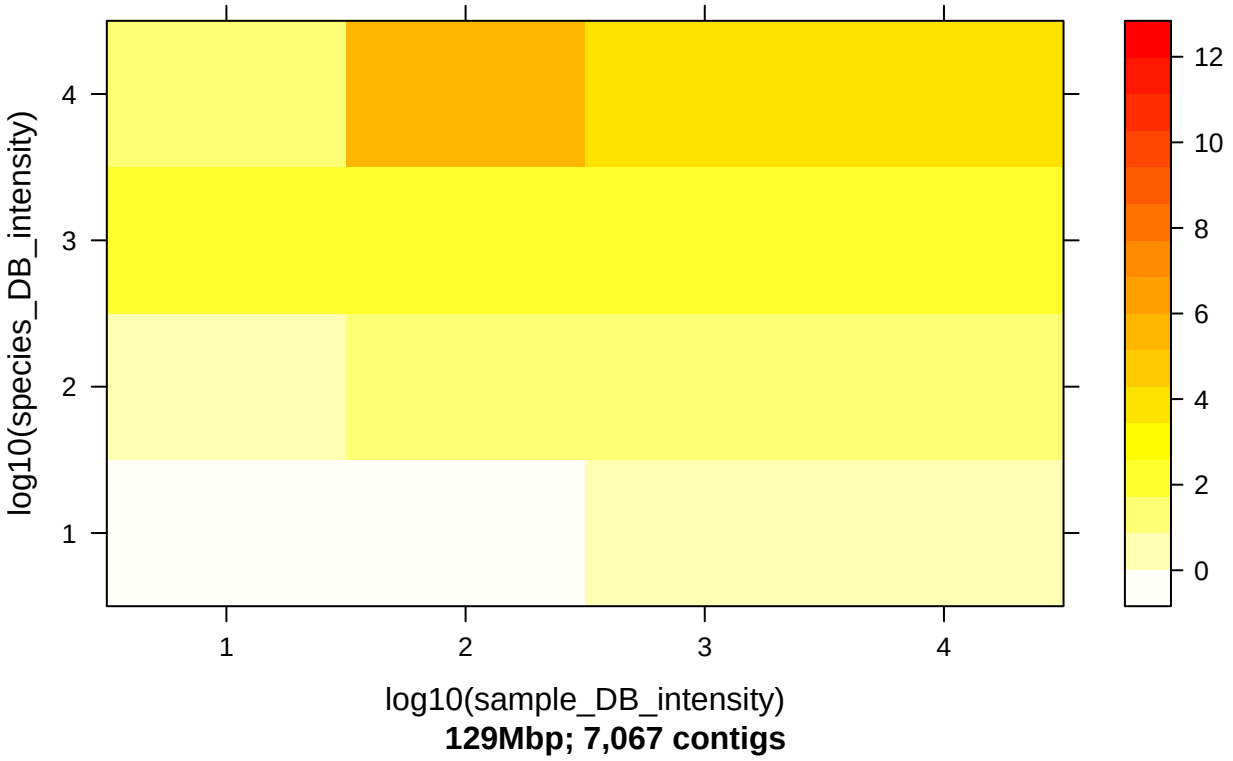
```
levelplot((one_way_TP + two_way_TP) / (one_way_FP + two_way_FP) ~ log10(sample_DB_intensity)*log10(spec
```



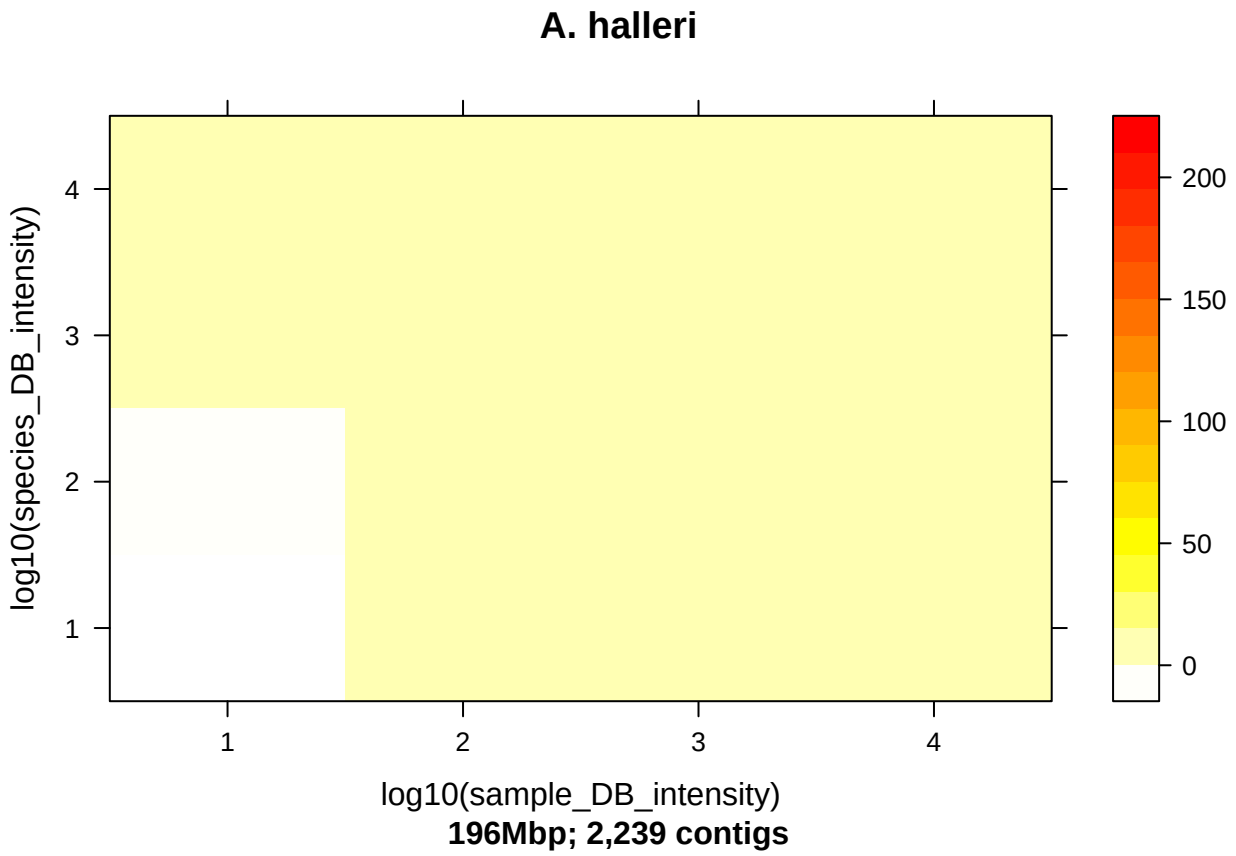

```
levelplot((one_way_TP + two_way_TP) / (one_way_FP + two_way_FP) ~ log10(sample_DB_intensity)*log10(spec
```



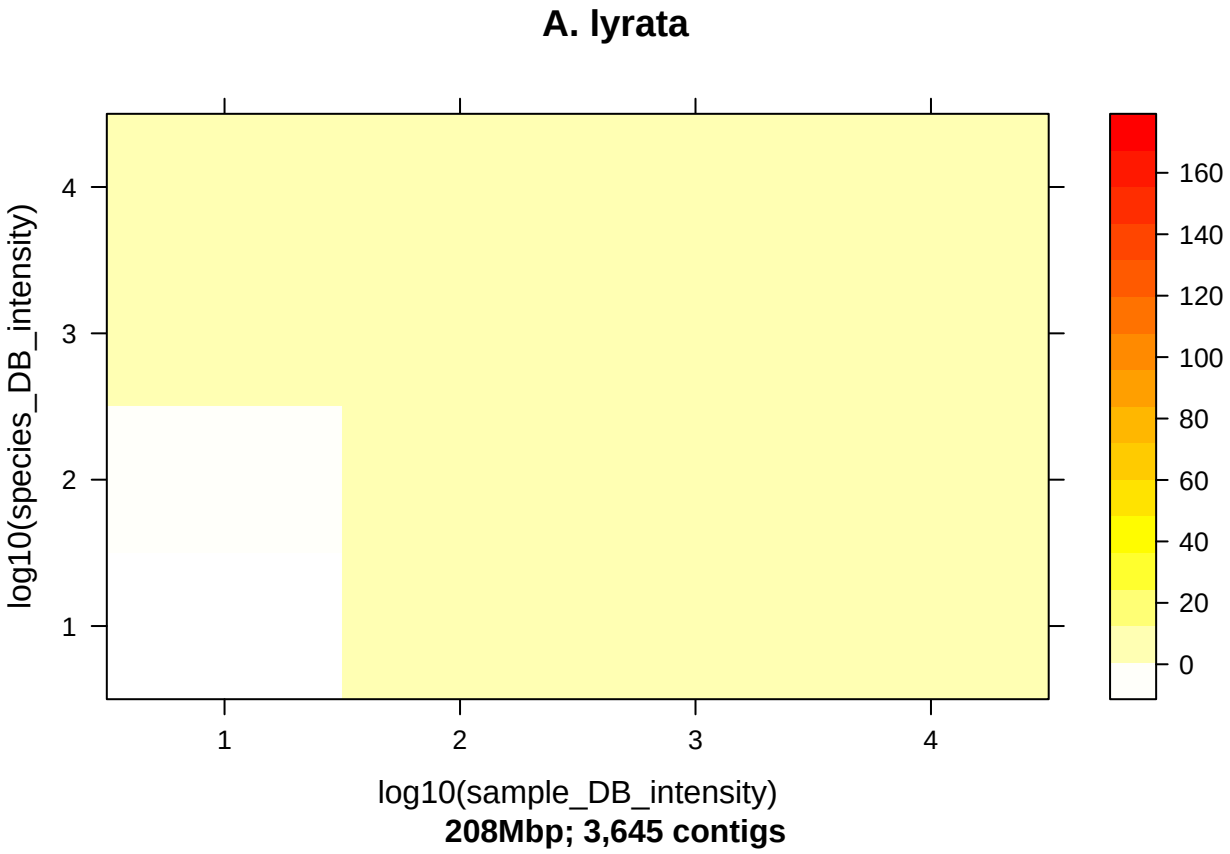

```
levelplot((one_way_TP + two_way_TP) / (one_way_FP + two_way_FP) ~ log10(sample_DB_intensity)*log10(spec
```



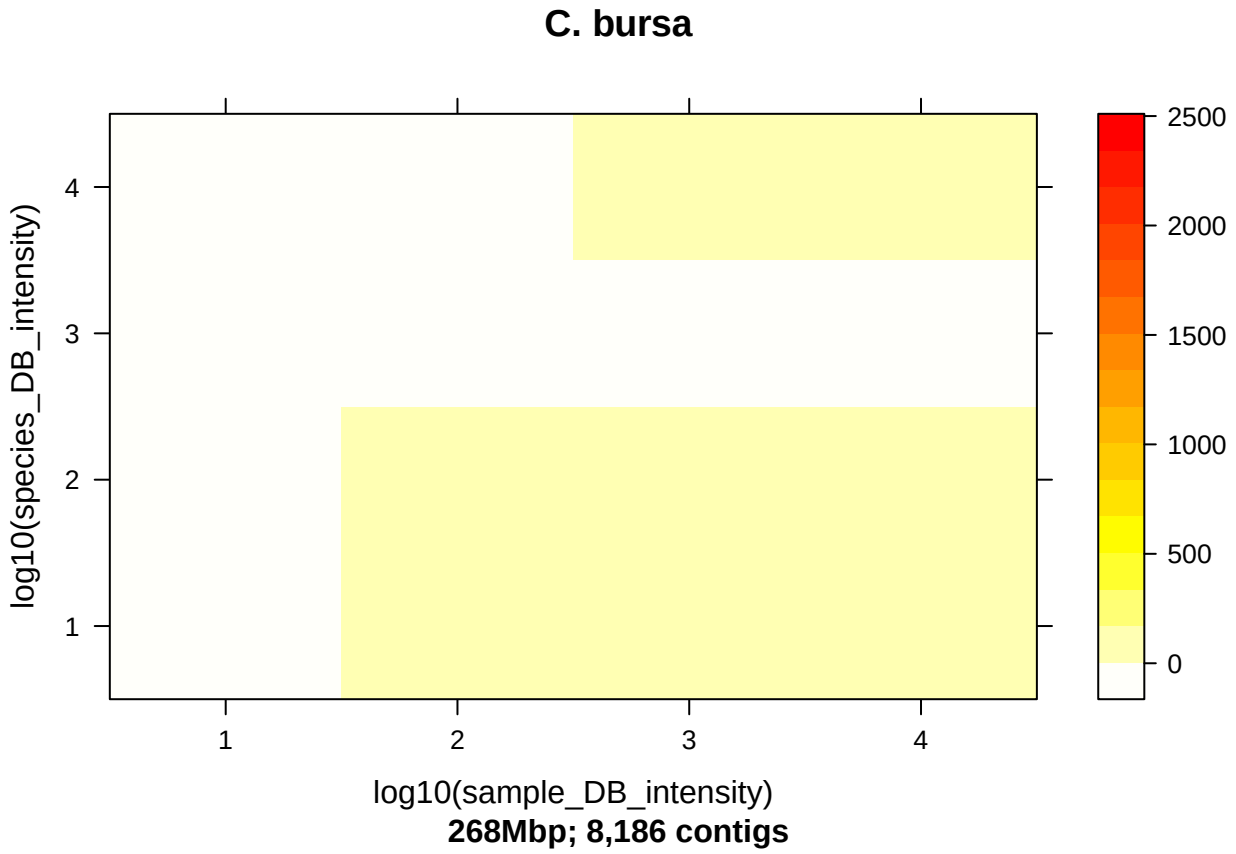

```
levelplot((one_way_TP + two_way_TP) / (one_way_FP + two_way_FP) ~ log10(sample_DB_intensity)*log10(spec
```



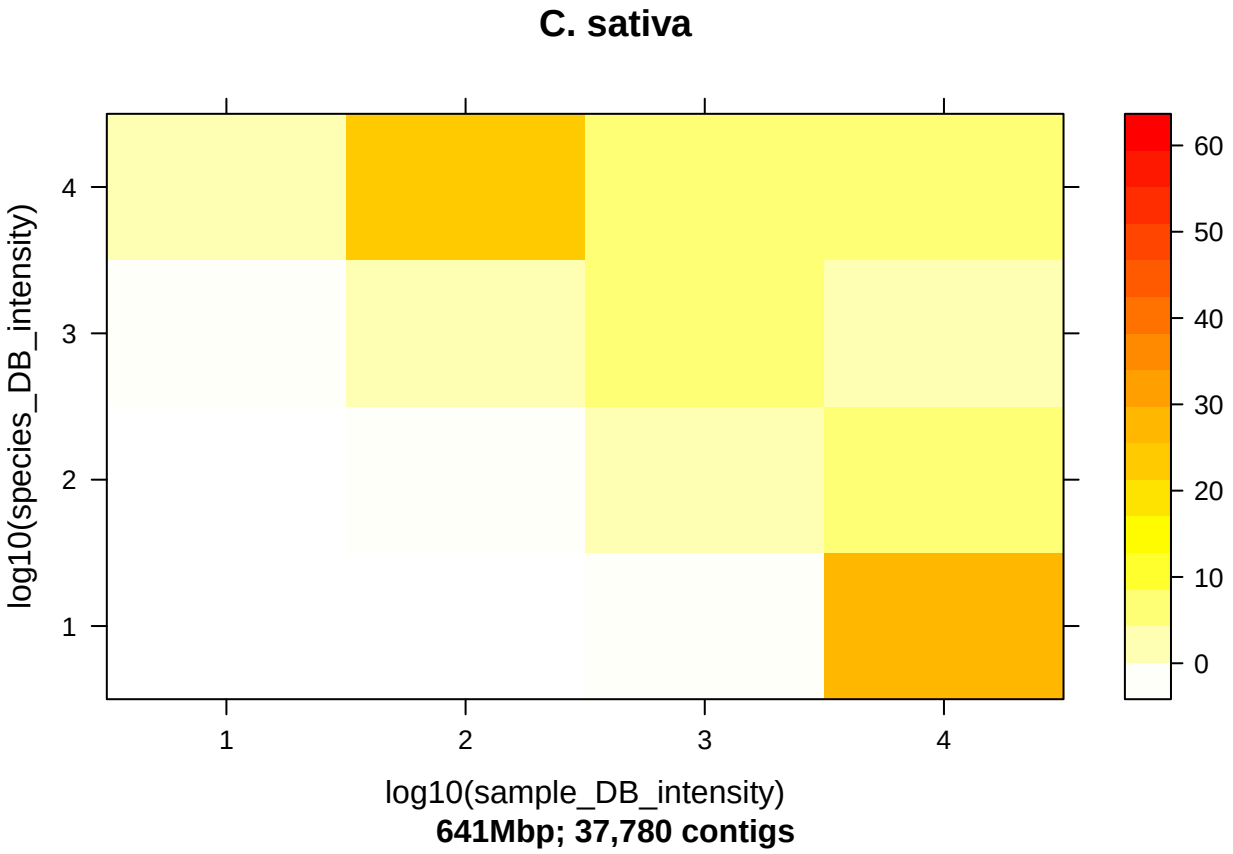

```
levelplot((one_way_TP + two_way_TP) / (one_way_FP + two_way_FP) ~ log10(sample_DB_intensity)*log10(spec
```




```
levelplot((one_way_TP + two_way_TP) / (one_way_FP + two_way_FP) ~ log10(sample_DB_intensity)*log10(spec
```

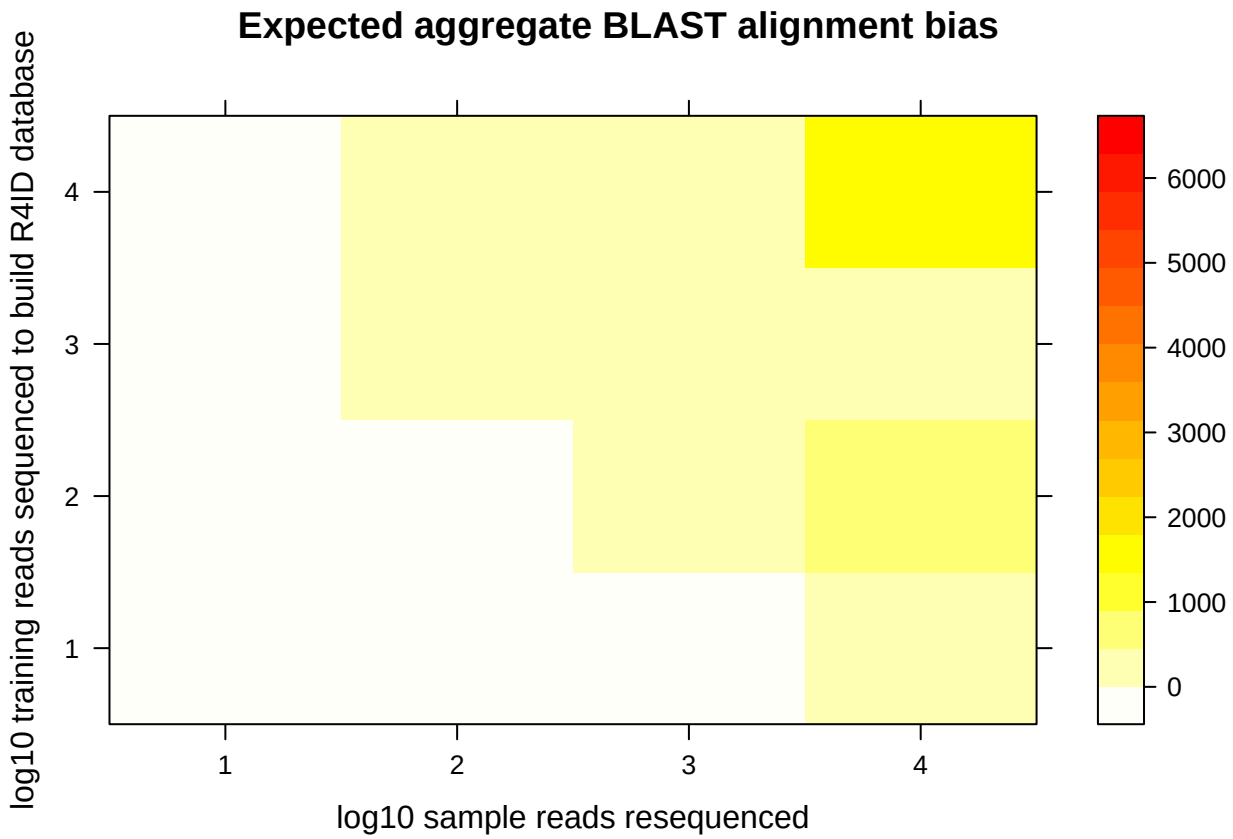



```
levelplot((one_way_TP + two_way_TP) / (one_way_FP + two_way_FP) ~ log10(sample_DB_intensity)*log10(spec
```



Thirdly *mean_bias*; first what is the aggregate length bias we can expect (as in the sci fest GUI)?

```
levelplot(two_way_rate*total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silic
```




```
# boxplots by intensity to see confidence intervals
par(mfrow=c(1,2))
boxplot(log10(two_way_rate*total_hits) ~ log10(sample_DB_intensity),data=in_silico, main="Expected aggr
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 1 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 2 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 3 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 4 is not drawn
```

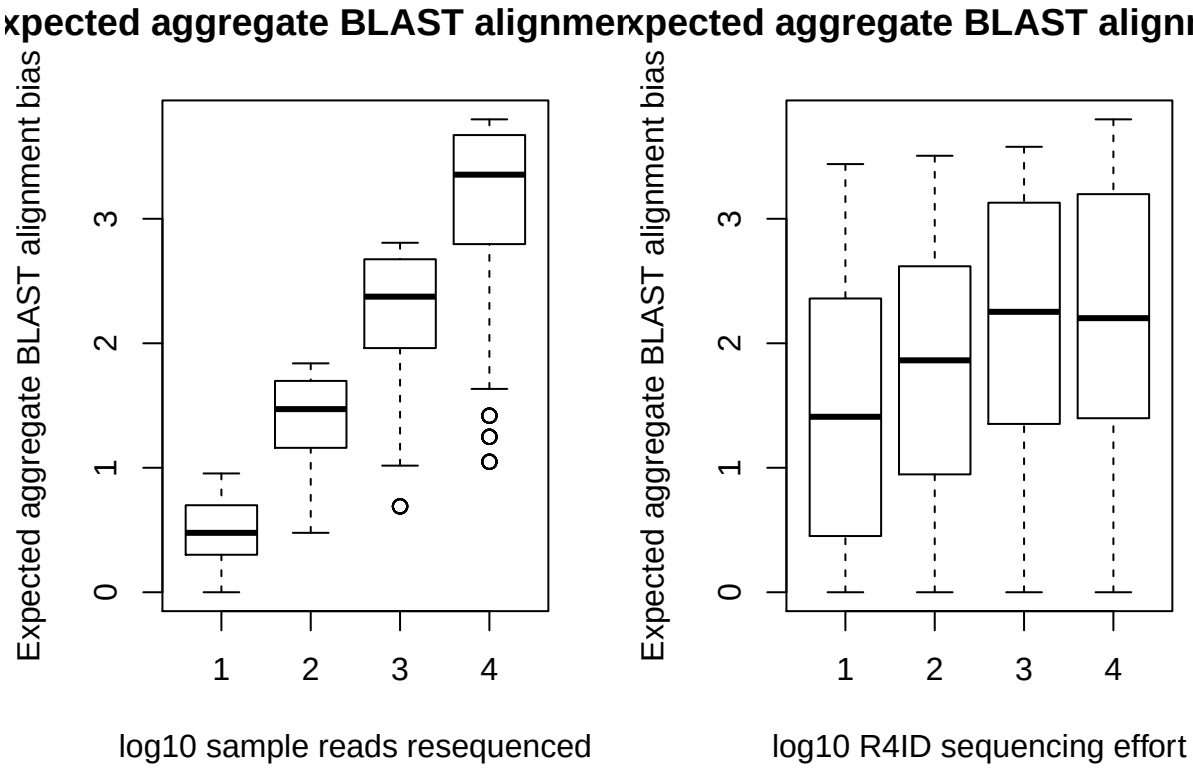
```
boxplot(log10(two_way_rate*total_hits) ~ log10(species_DB_intensity),data=in_silico, main="Expected agg
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 1 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 2 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 3 is not drawn
```

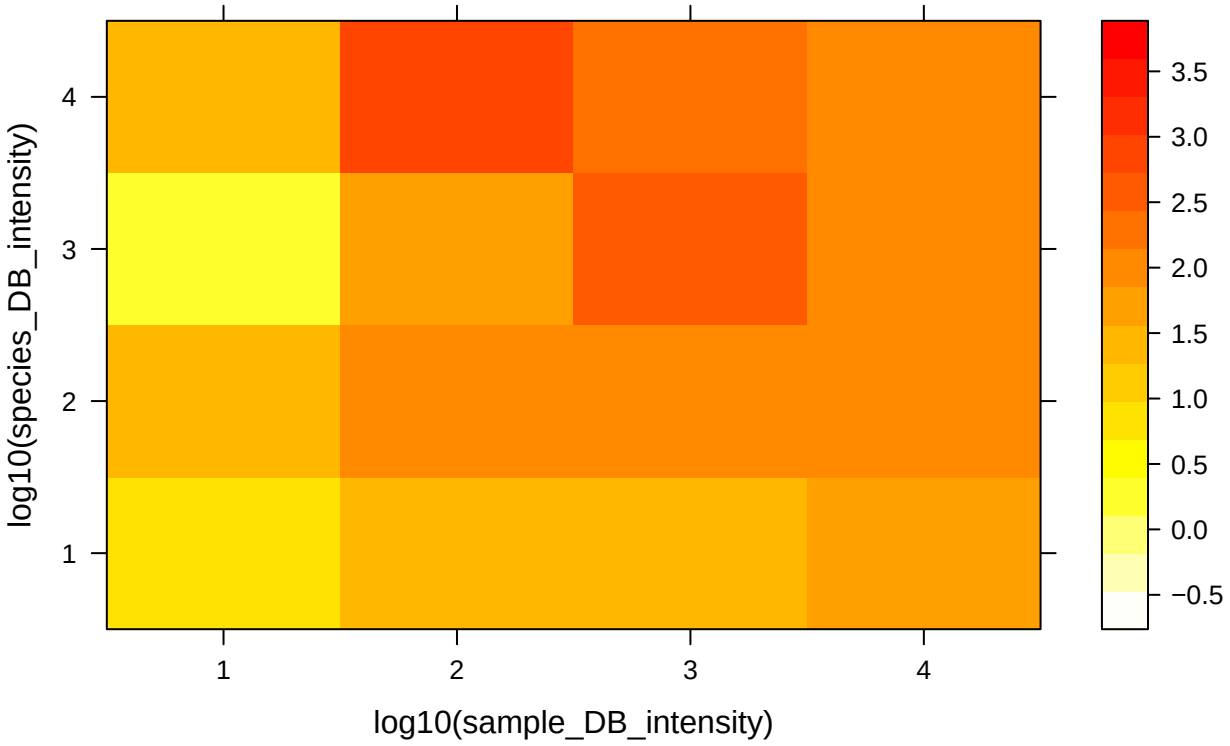
```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 4 is not drawn
```



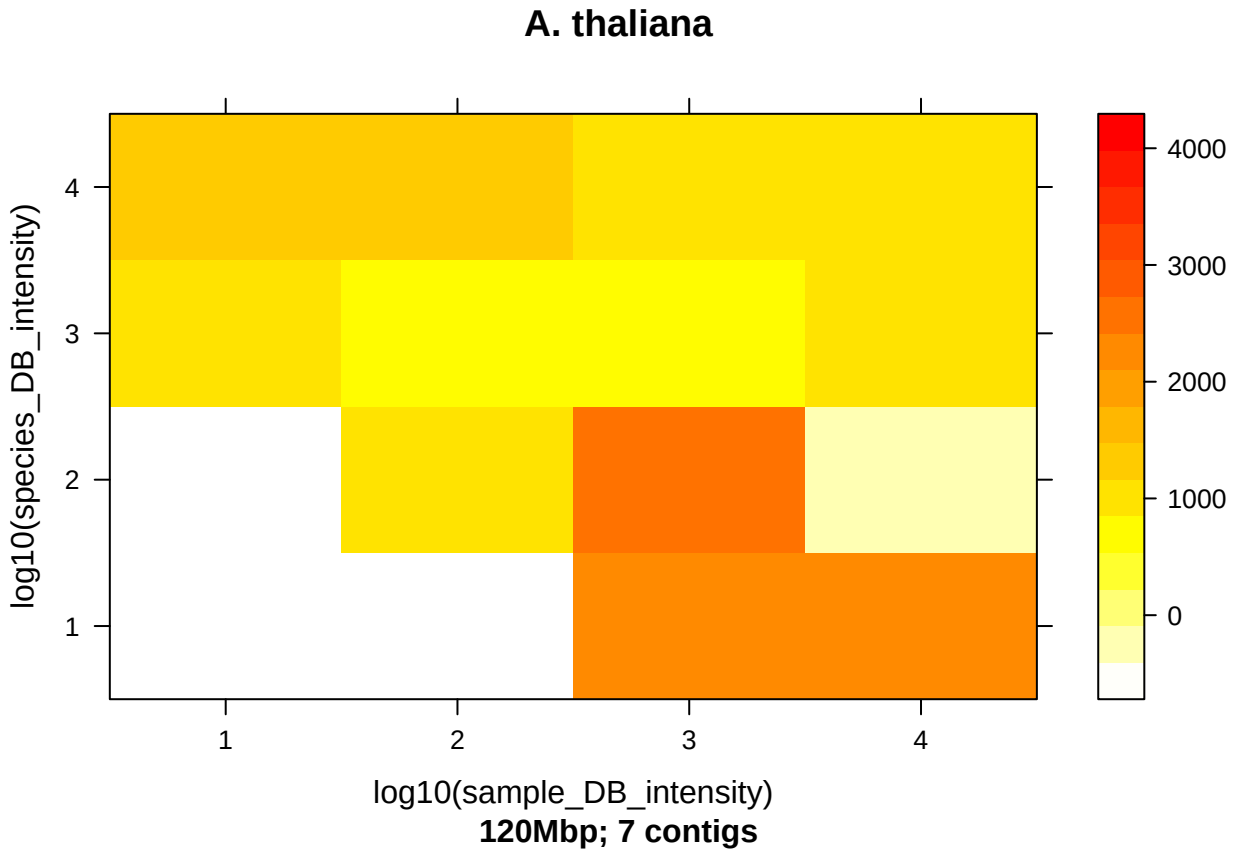
And the bias itself?

```
levelplot(log10(mean_bias) ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico, col
```

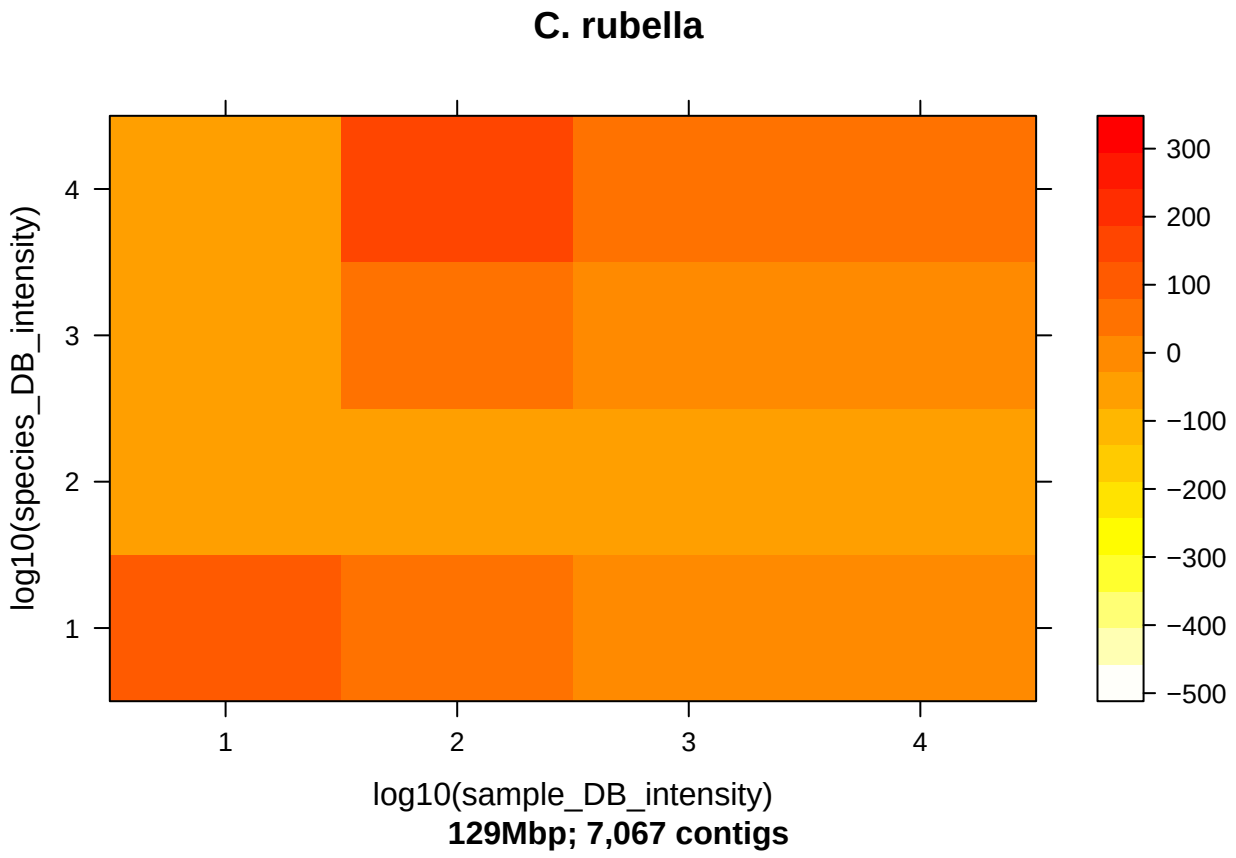
```
## Warning in eval(i, data, env): NaNs produced
```



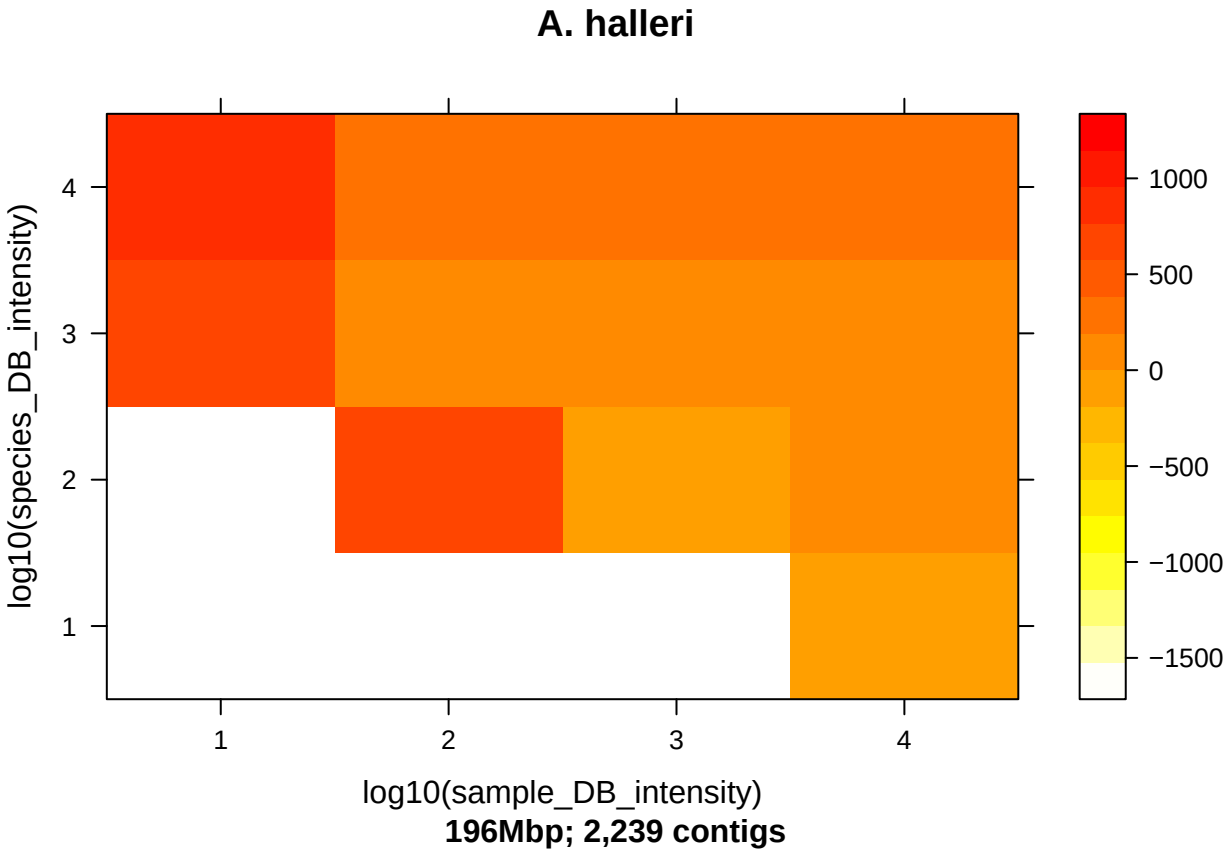

```
levelplot(mean_bias ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$T
```



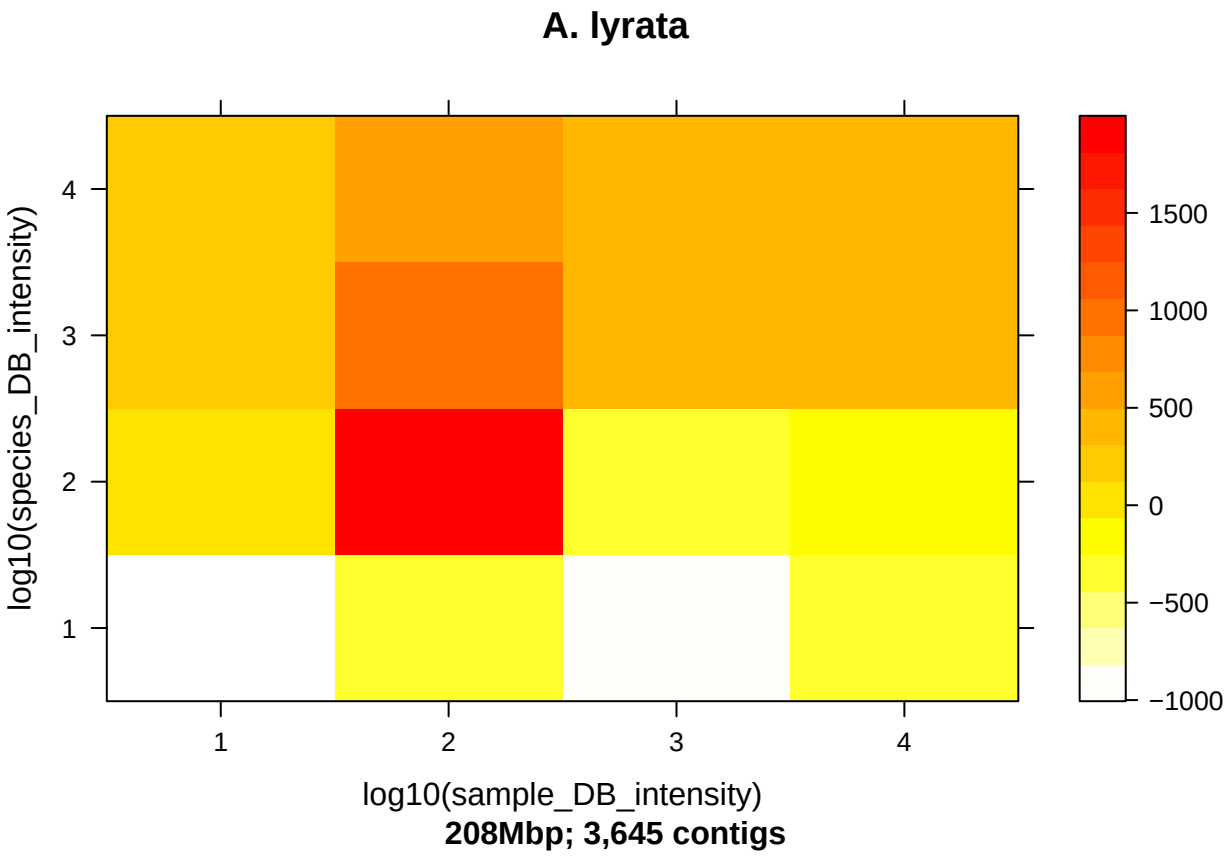

```
levelplot(mean_bias ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$T
```



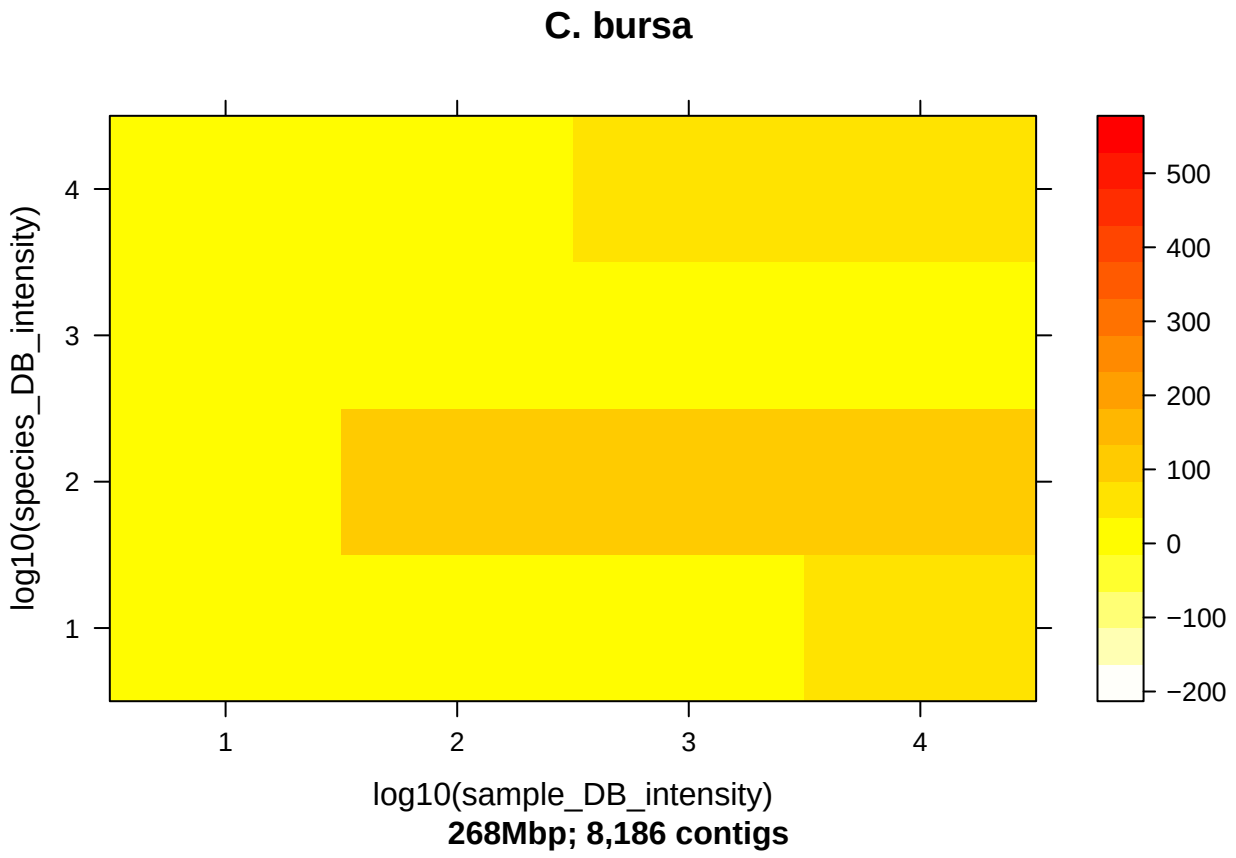

```
levelplot(mean_bias ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$T
```



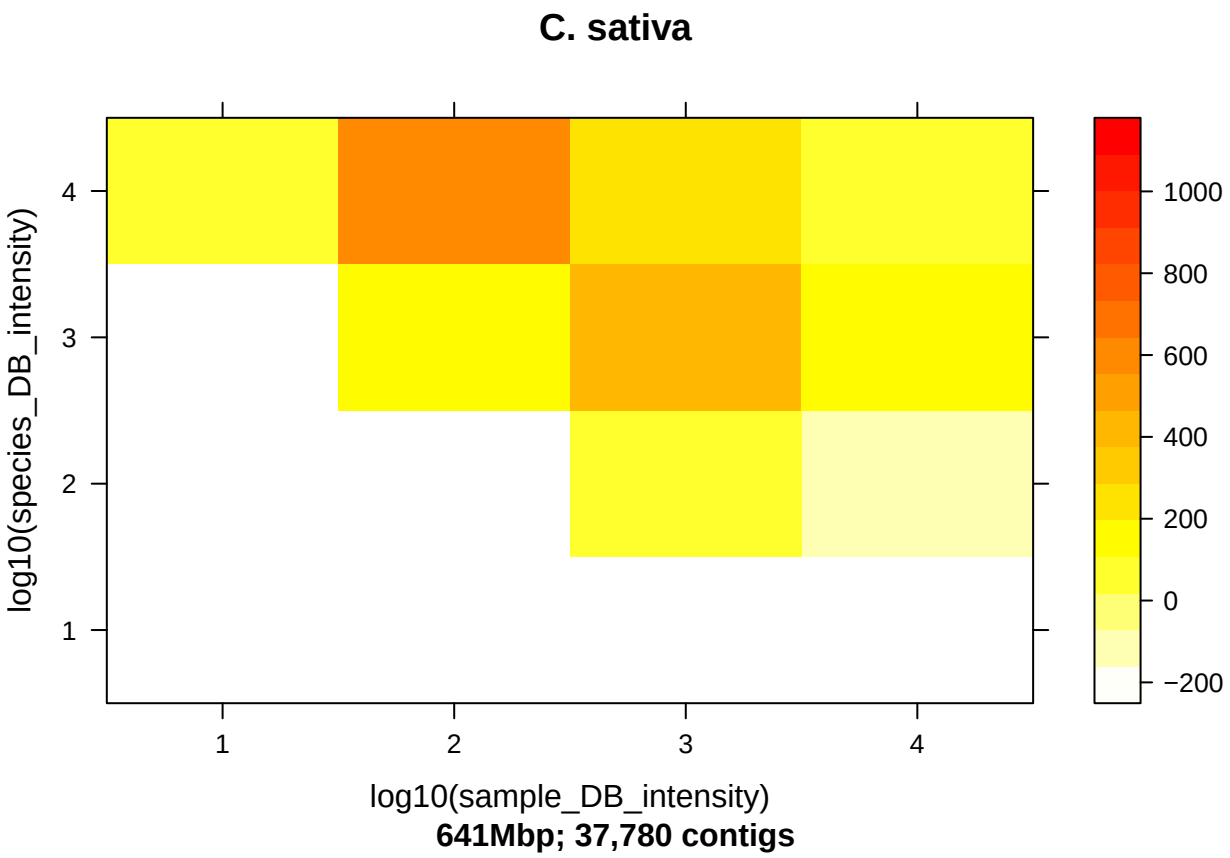

```
levelplot(mean_bias ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$T
```




```
levelplot(mean_bias ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$T
```

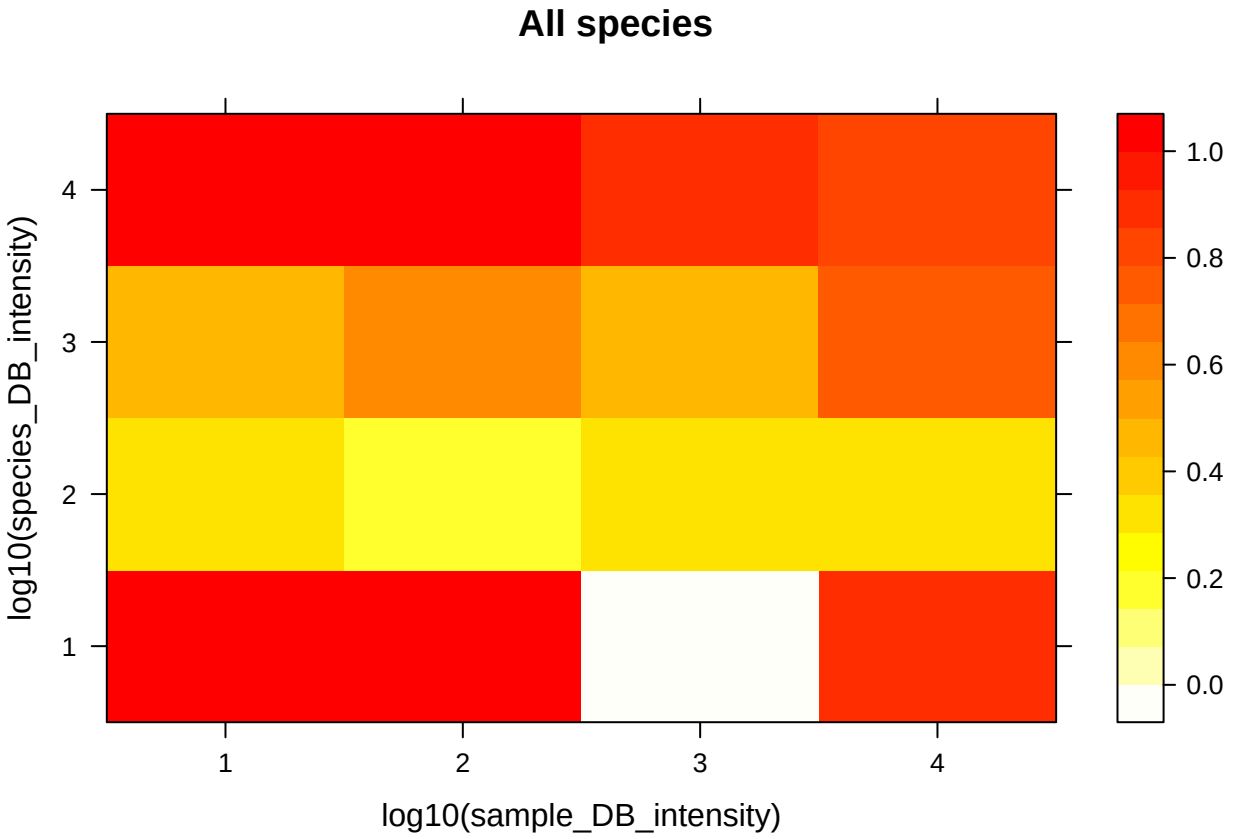



```
levelplot(mean_bias ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$TI
```

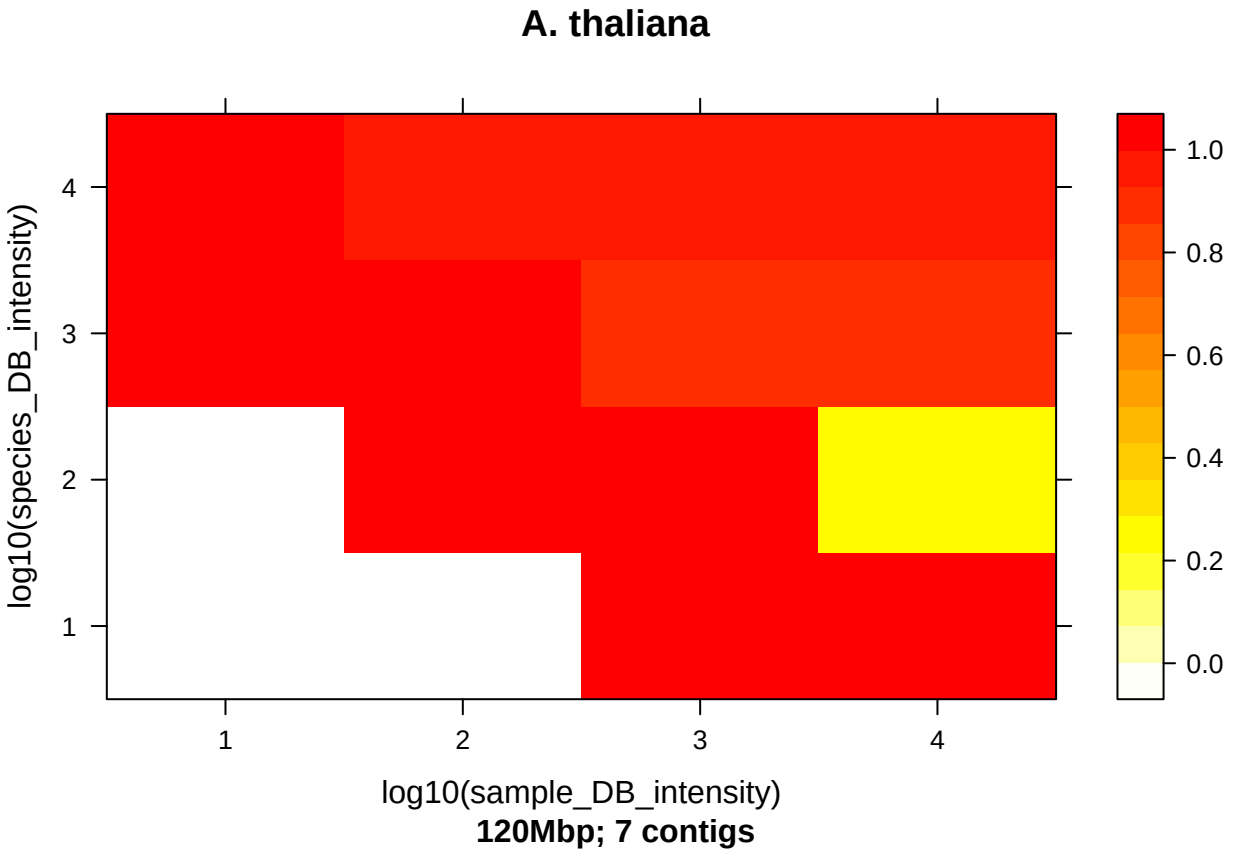


Fourth *hit rate in 2-way assignments*

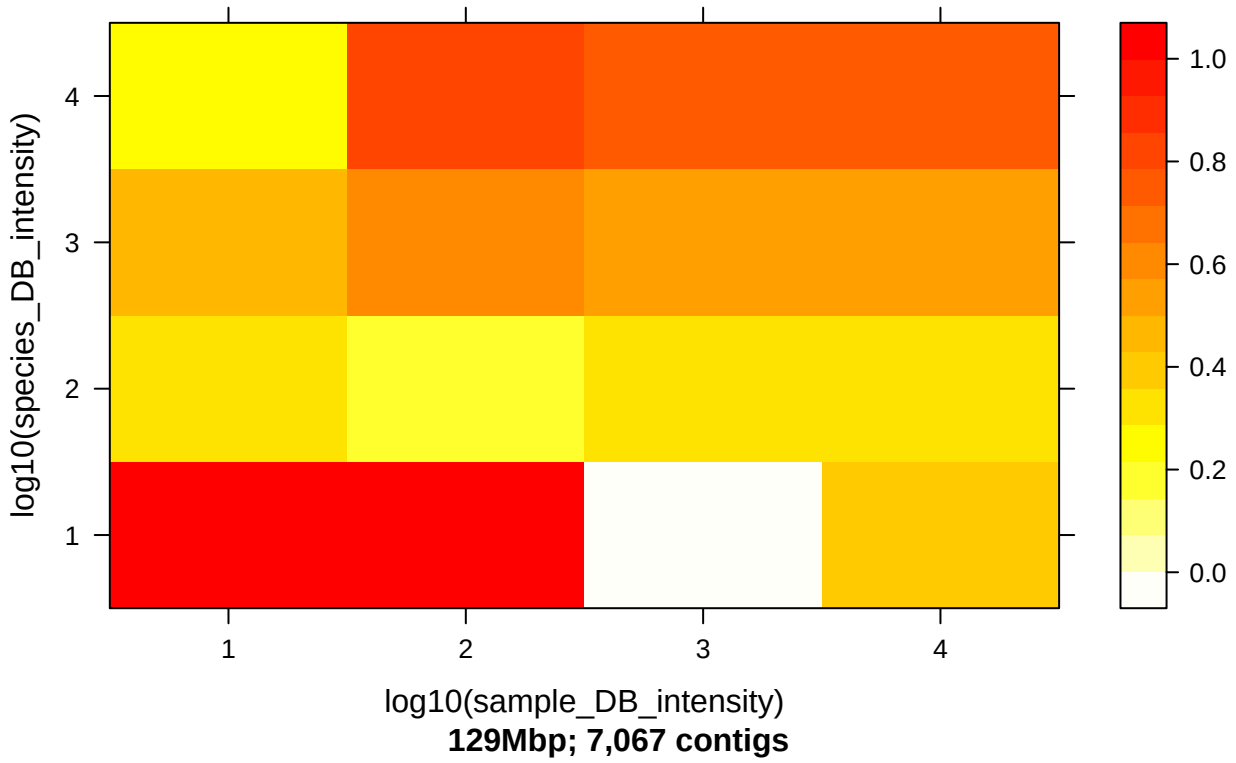
```
levelplot(two_way_rate ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico, col.reg
```



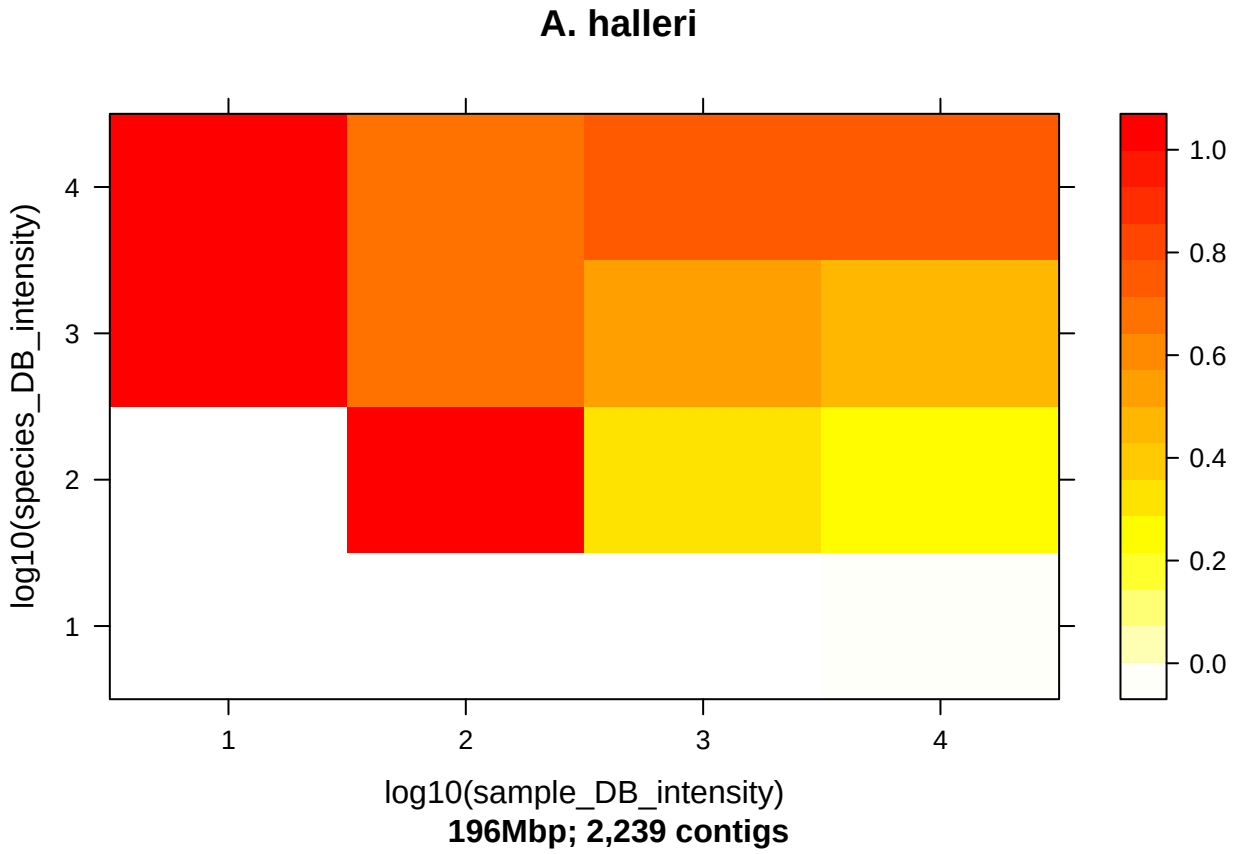

```
levelplot(two_way_rate ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silic
```



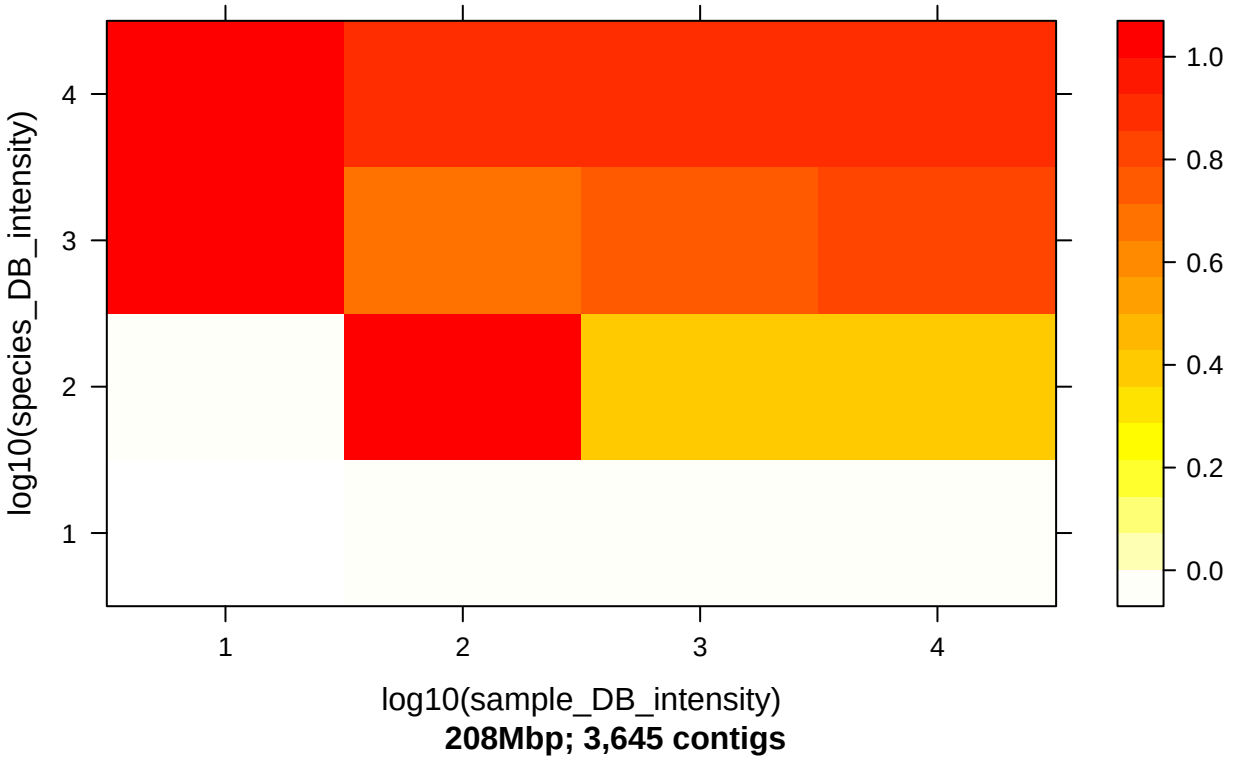

```
levelplot(two_way_rate ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silic
```



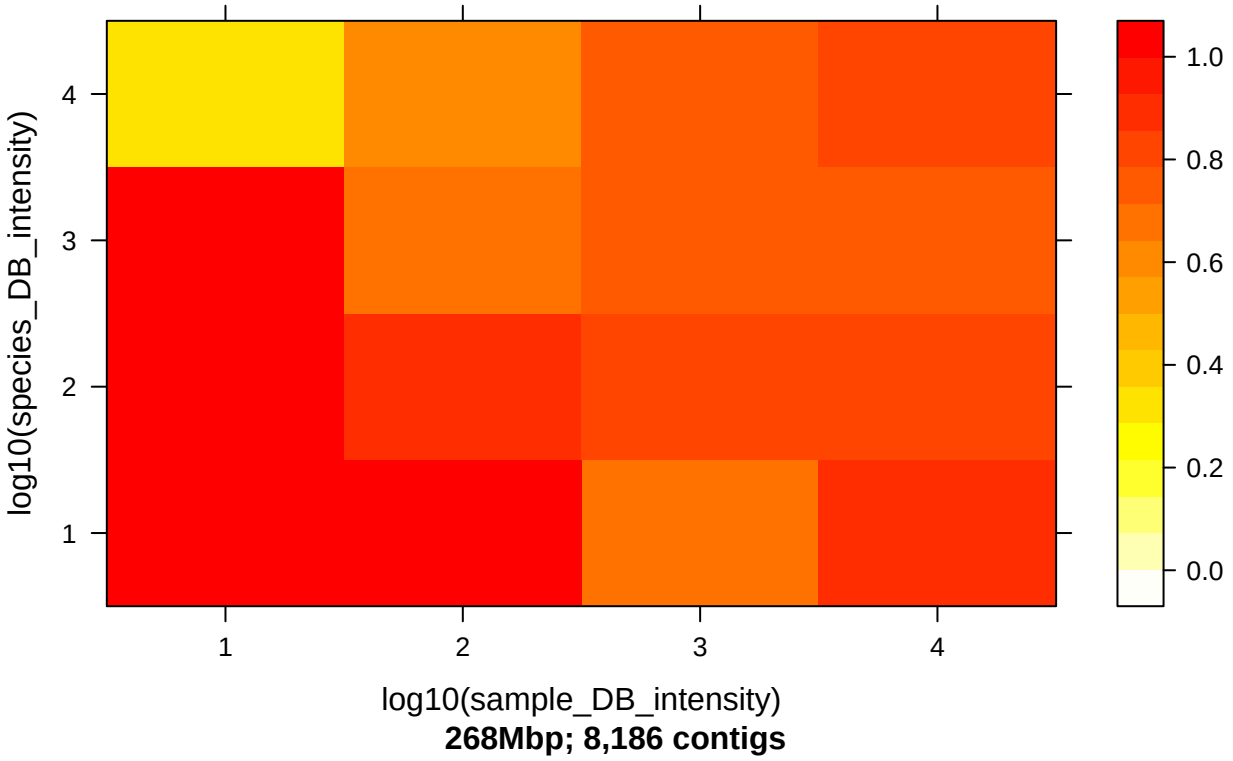

```
levelplot(two_way_rate ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silic
```



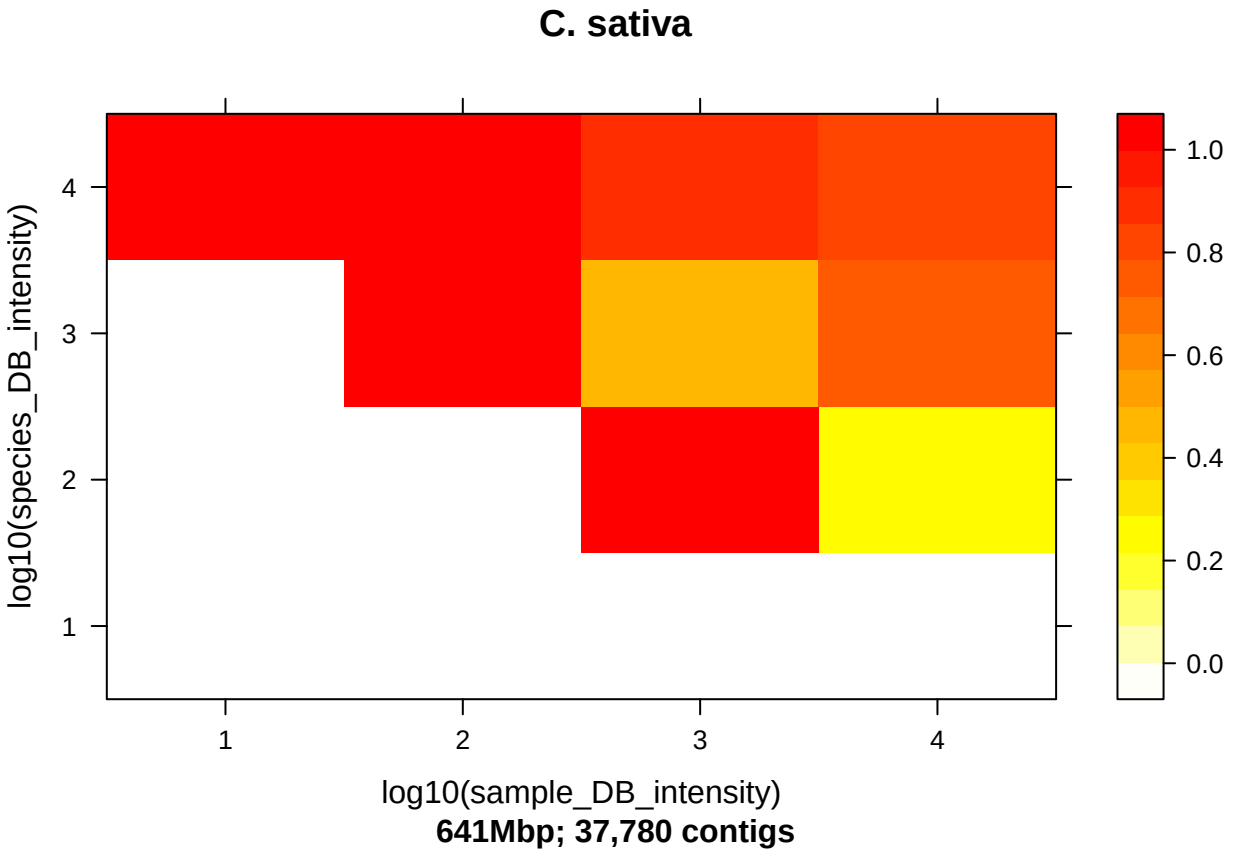

```
levelplot(two_way_rate ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silic
```




```
levelplot(two_way_rate ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silic
```

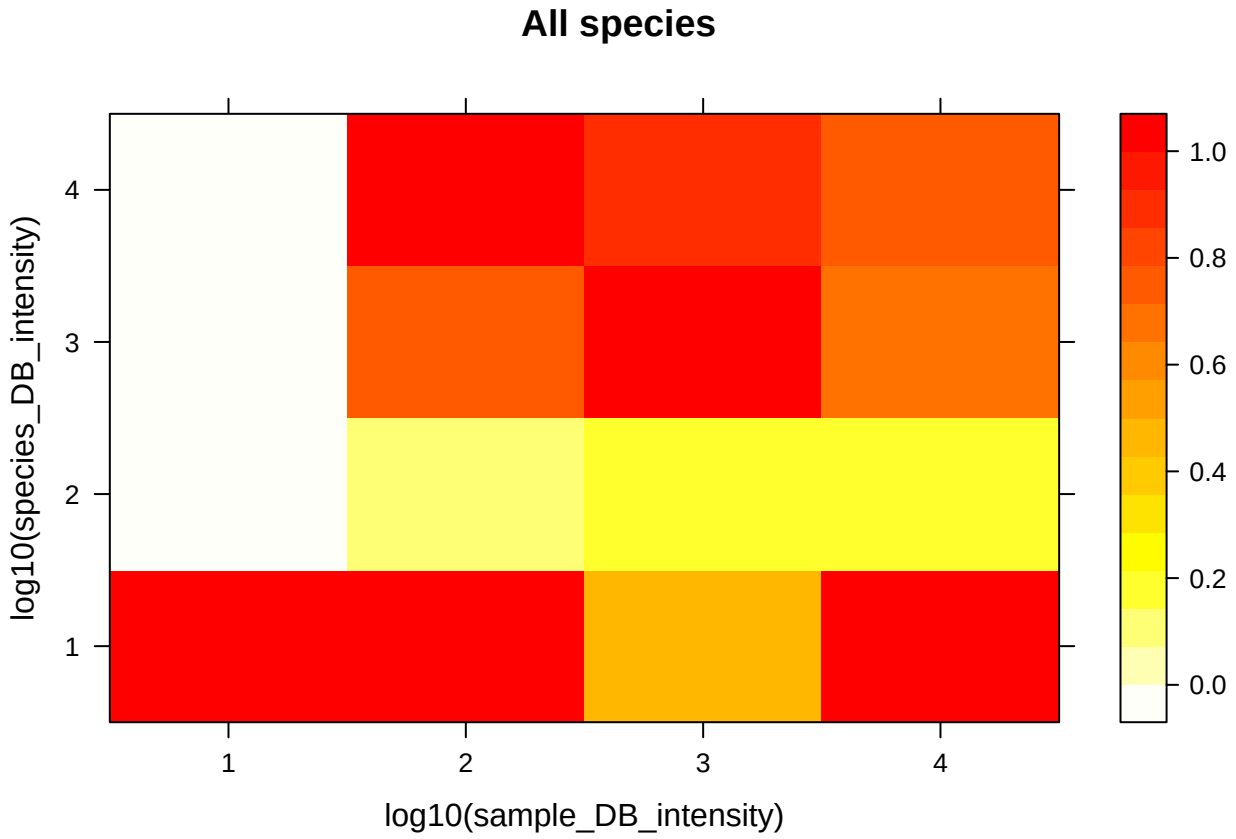



```
levelplot(two_way_rate ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silic
```

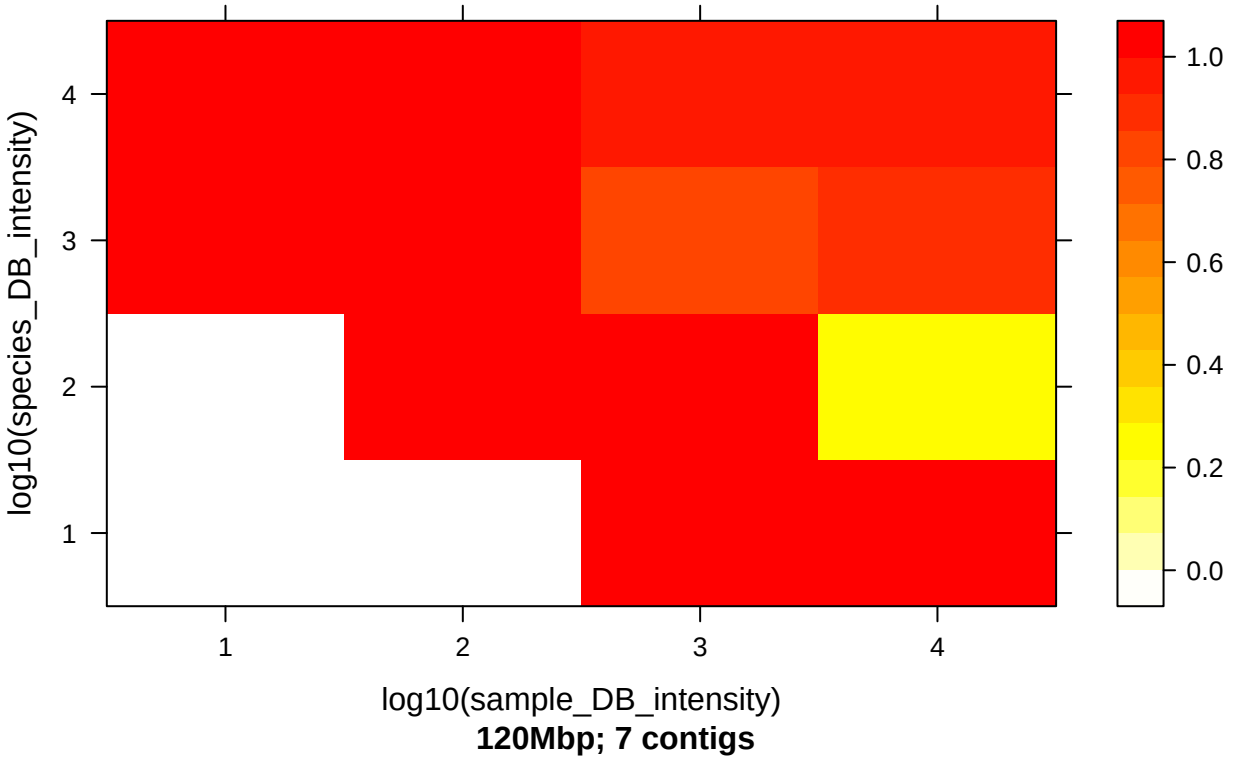


Fifth *hit rate in 2-way assignments, with cutoff>50*

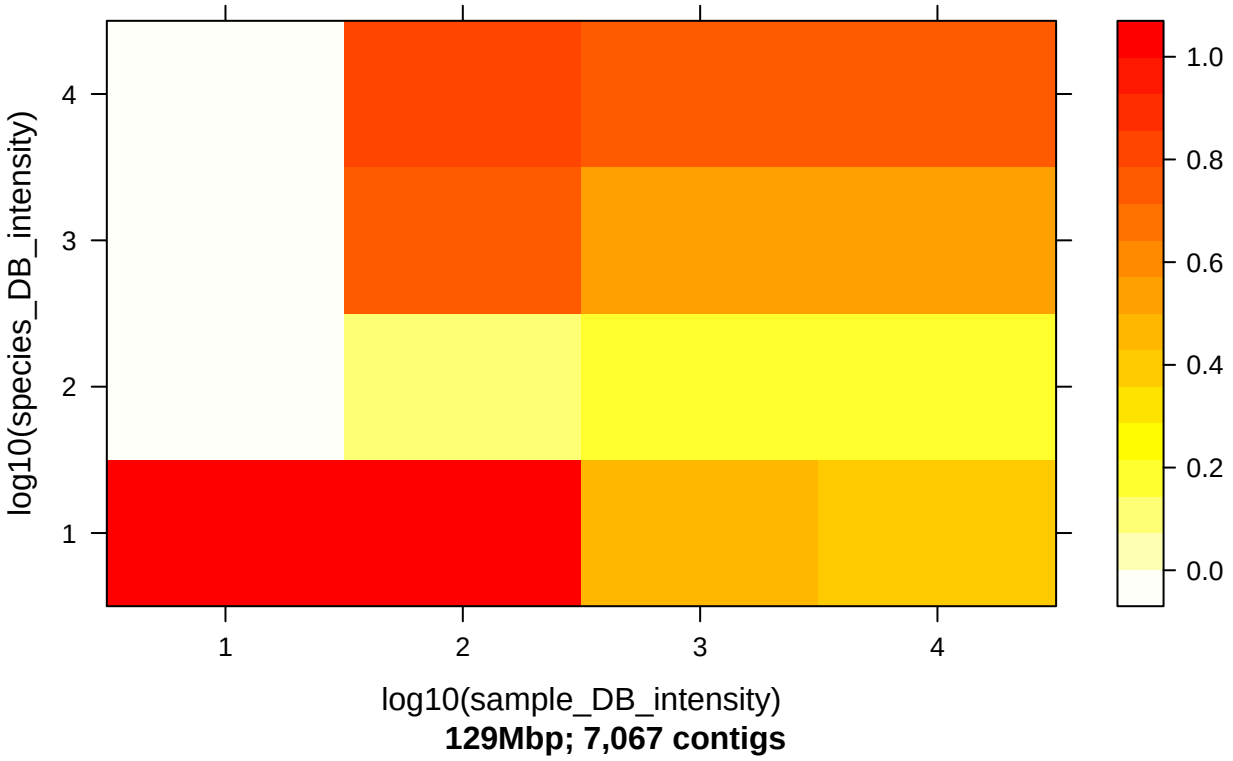
```
levelplot(two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_sil
```



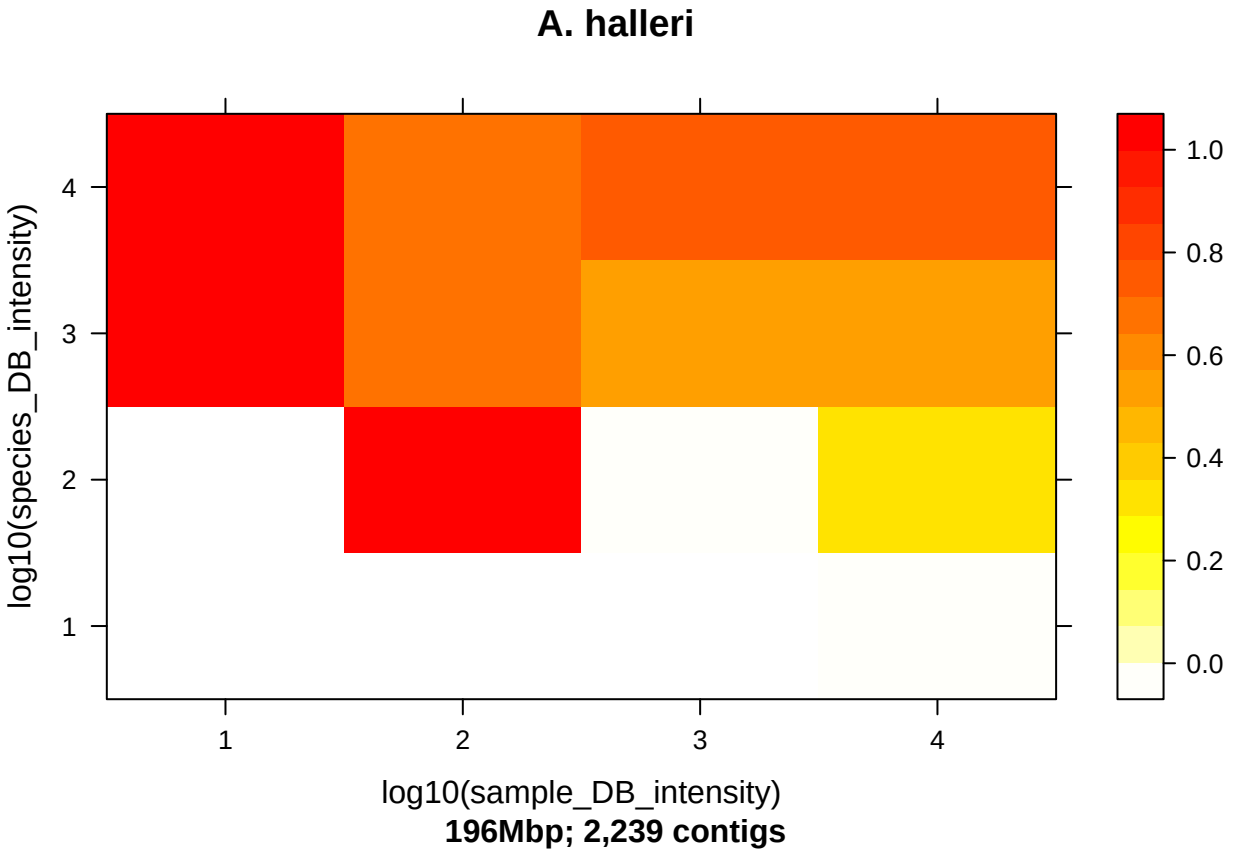

```
levelplot(two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_sil
```



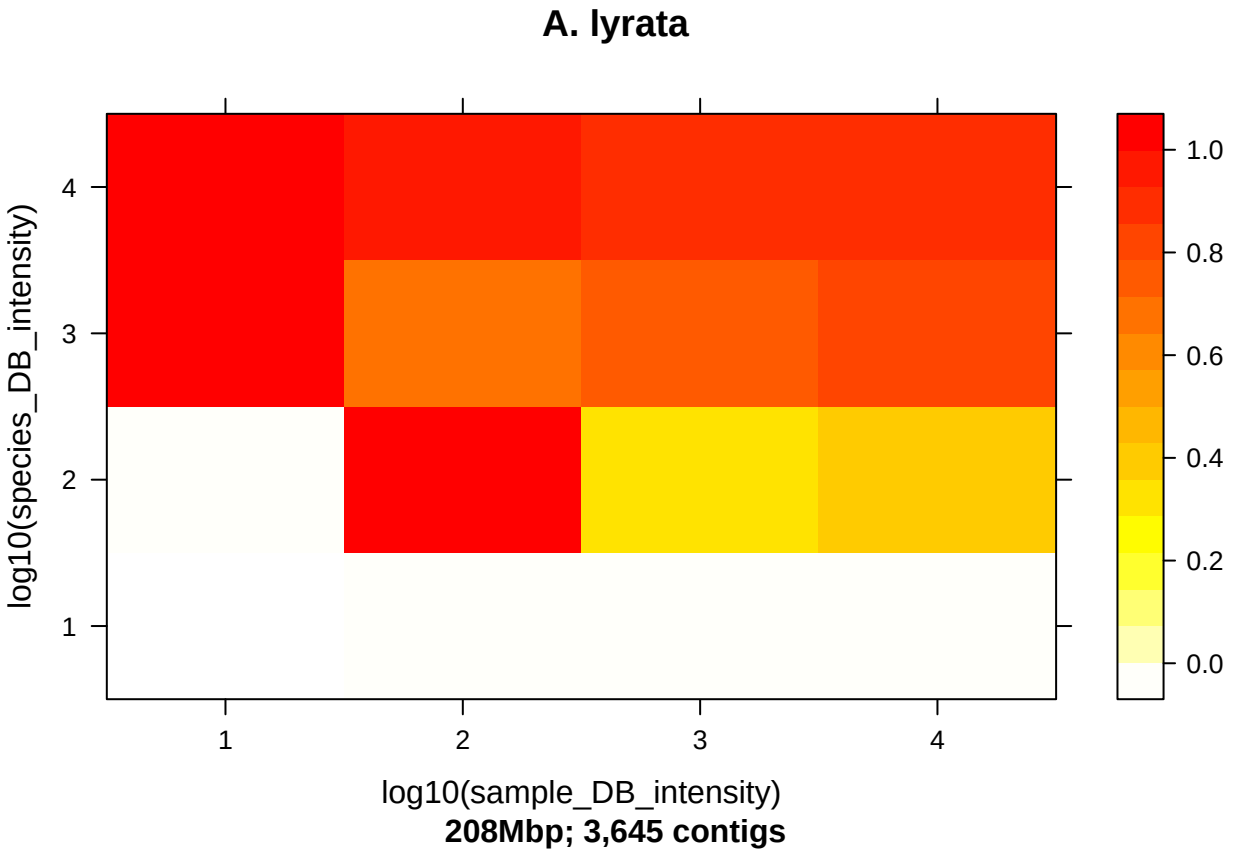

```
levelplot(two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_sil
```



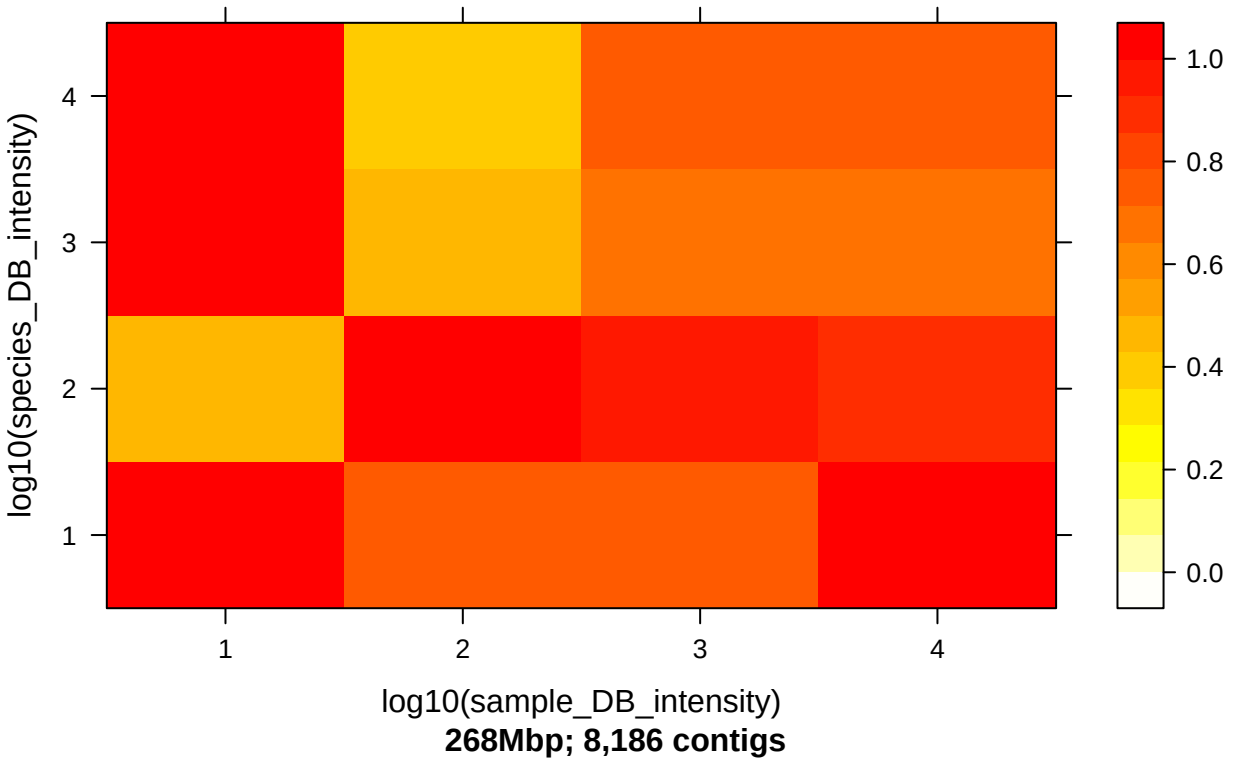

```
levelplot(two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_sil
```



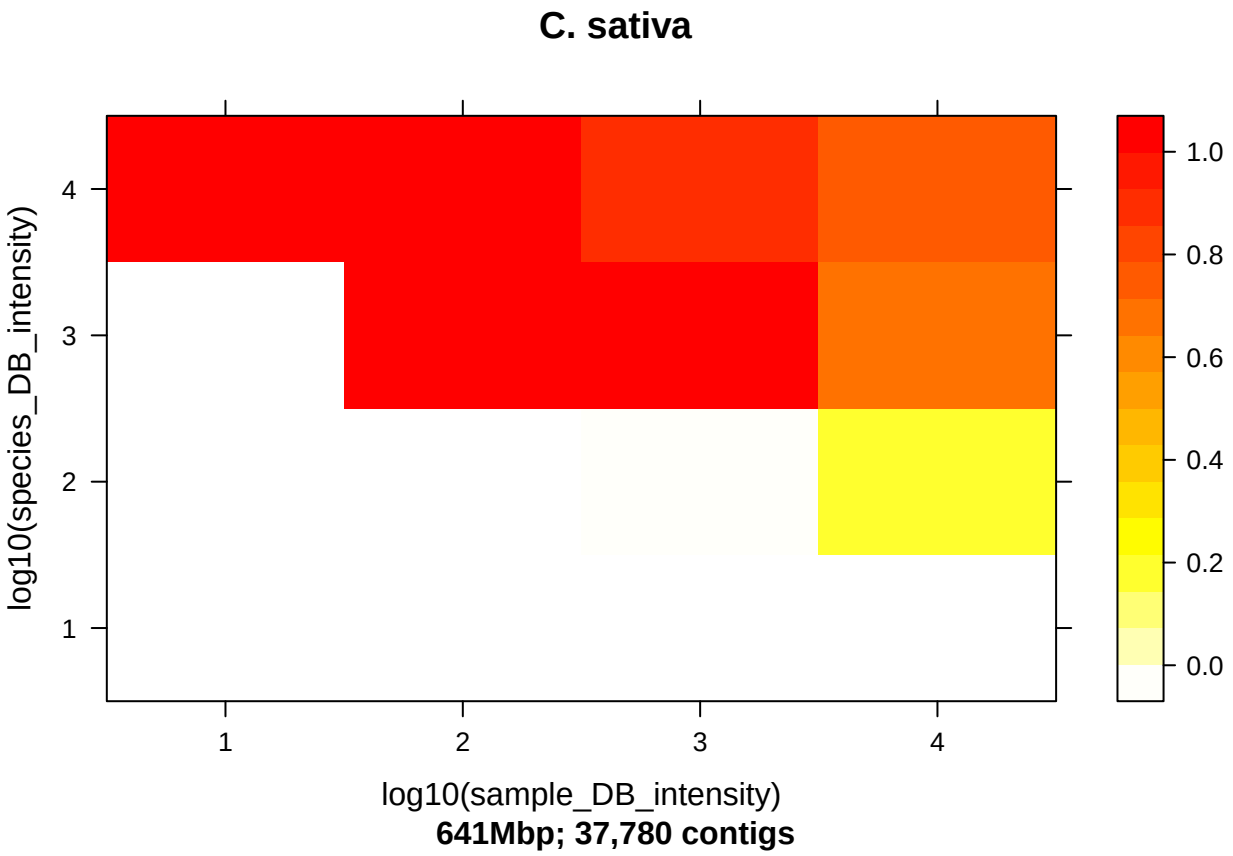

```
levelplot(two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_sil
```




```
levelplot(two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_sil
```

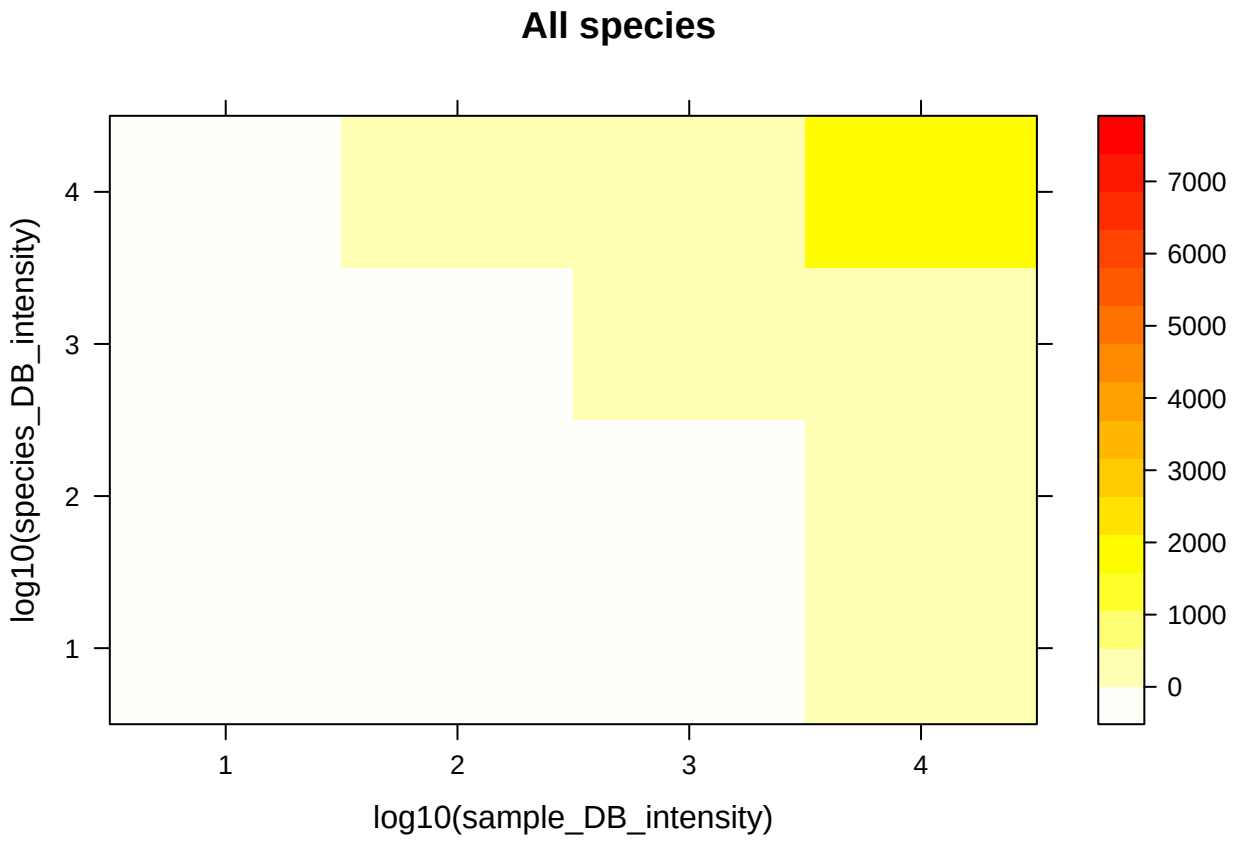



```
levelplot(two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_sil
```

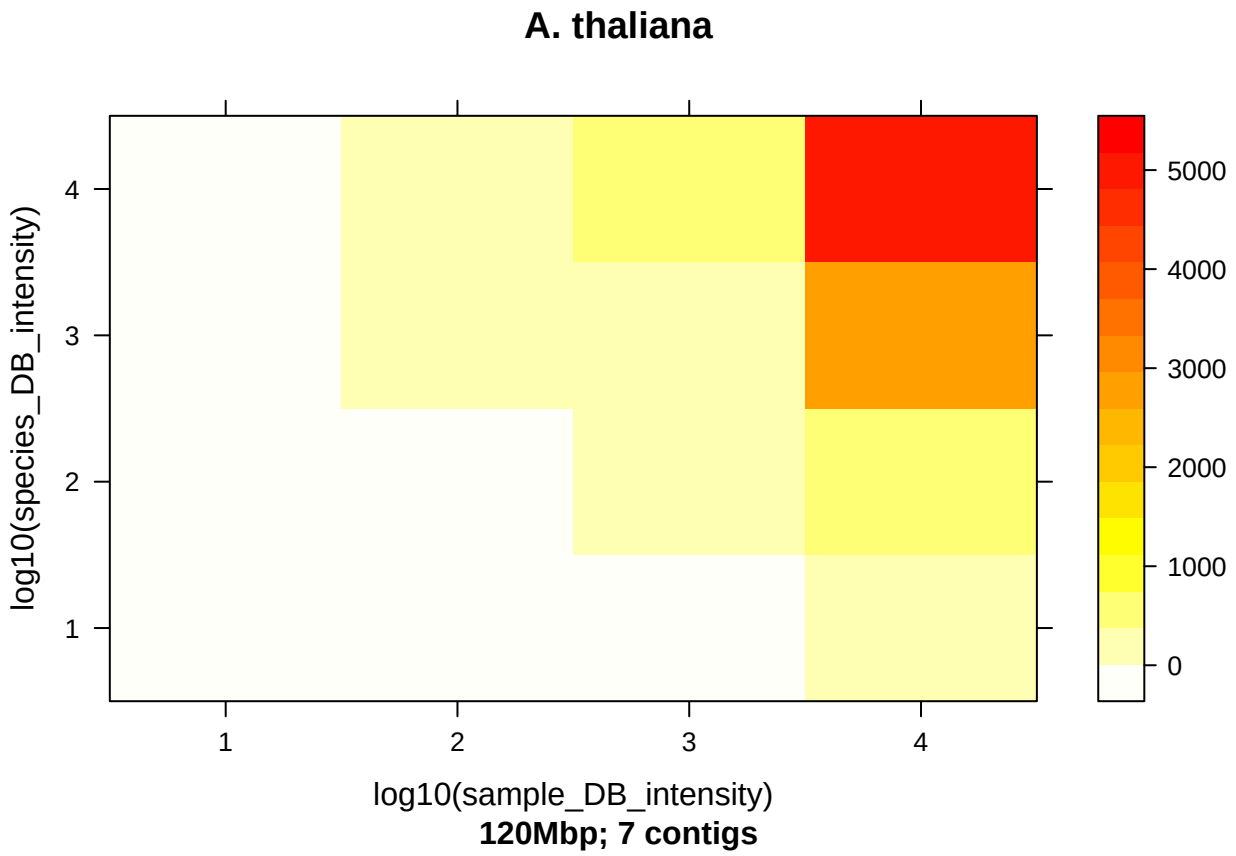


Finally *total hits*

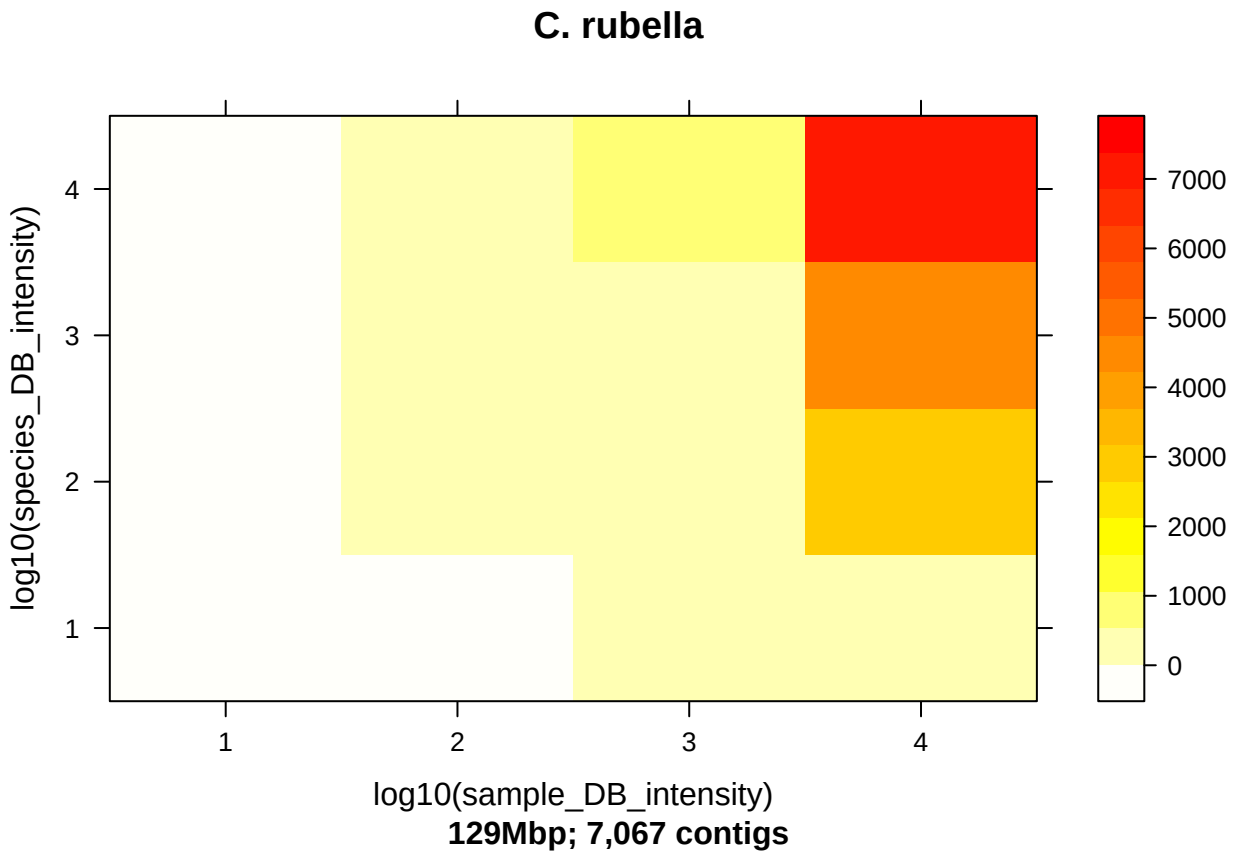
```
levelplot(total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico, col.regio
```



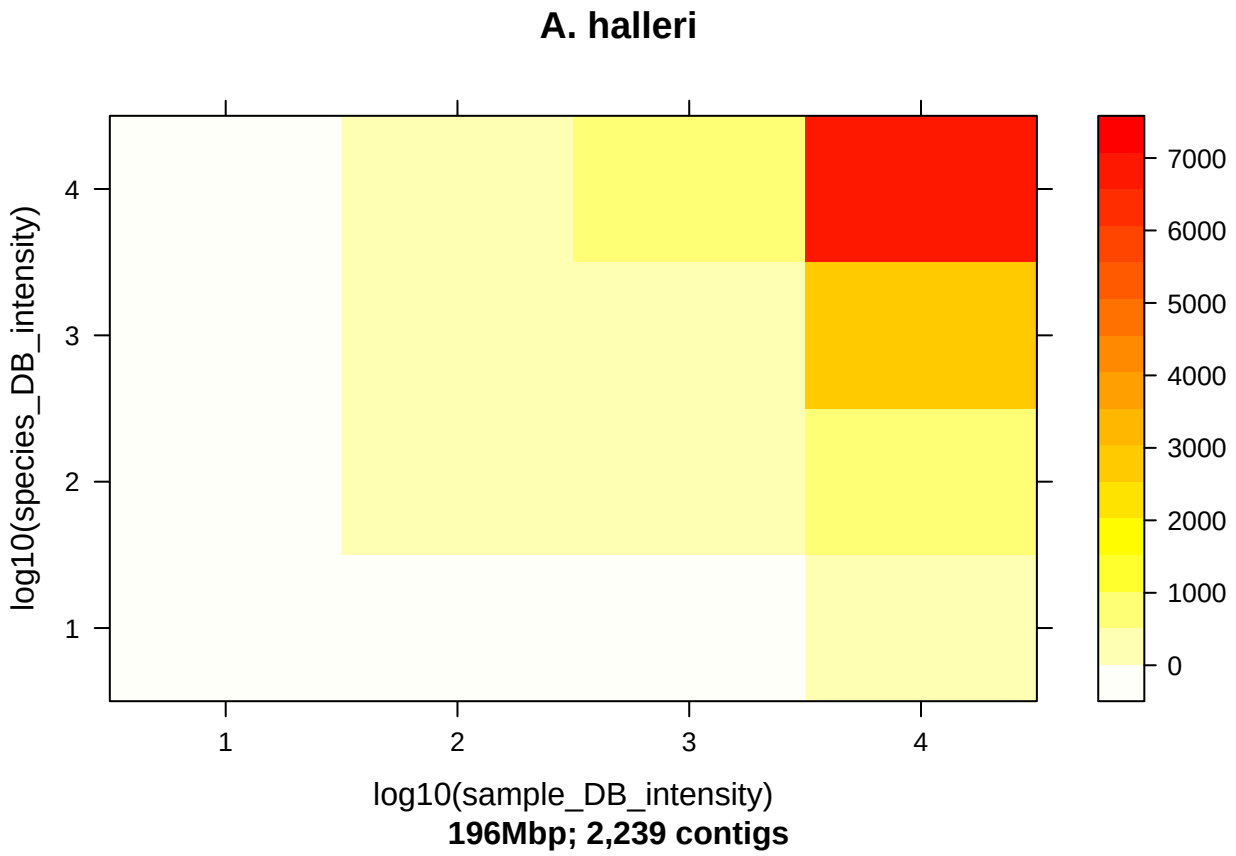

```
levelplot(total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$
```



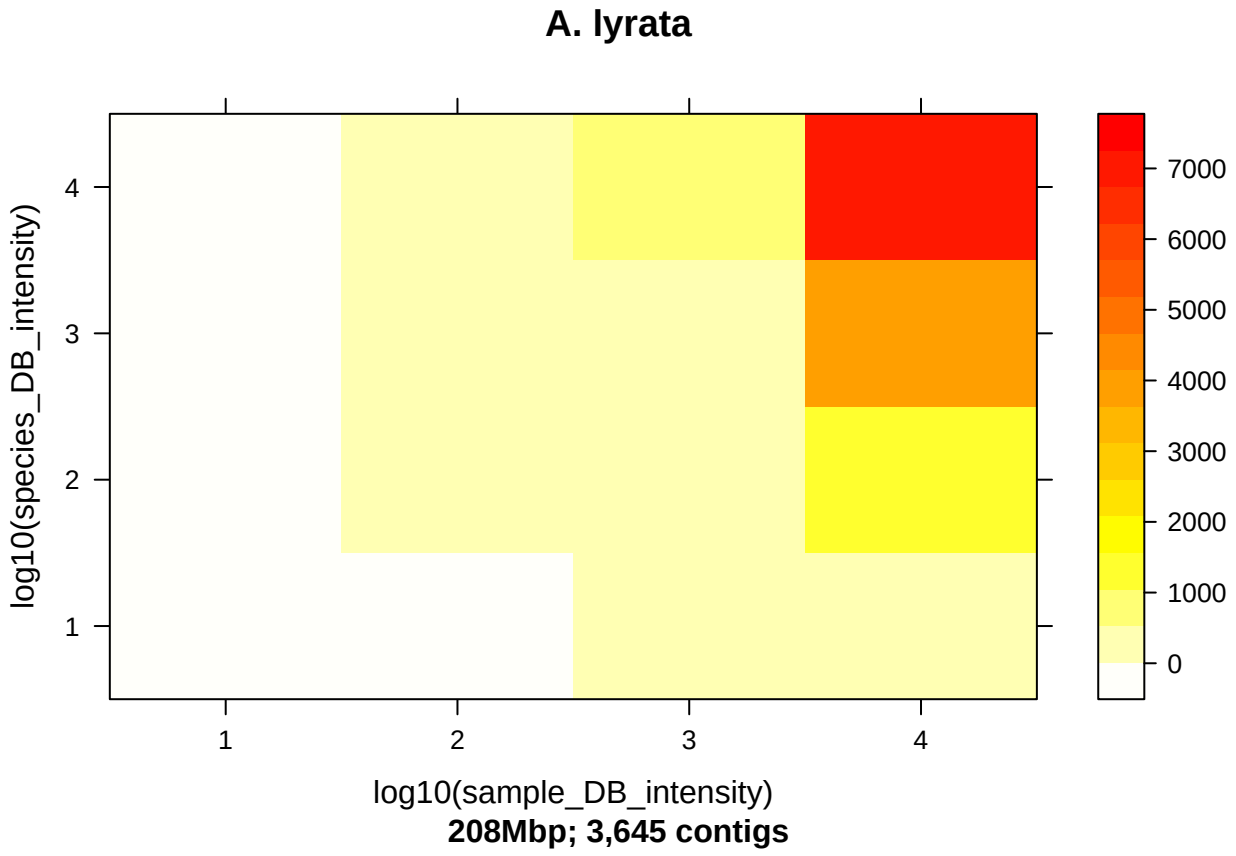

```
levelplot(total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$
```



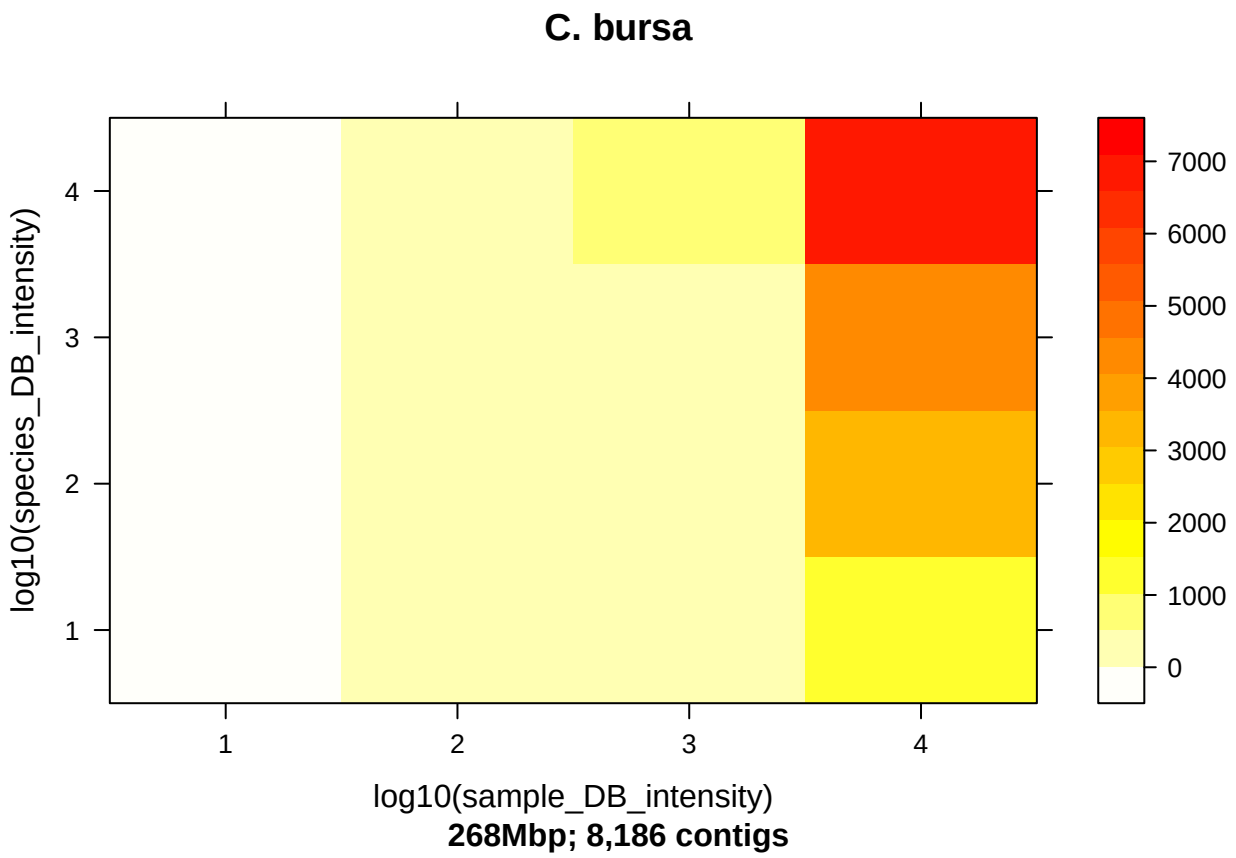

```
levelplot(total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$
```



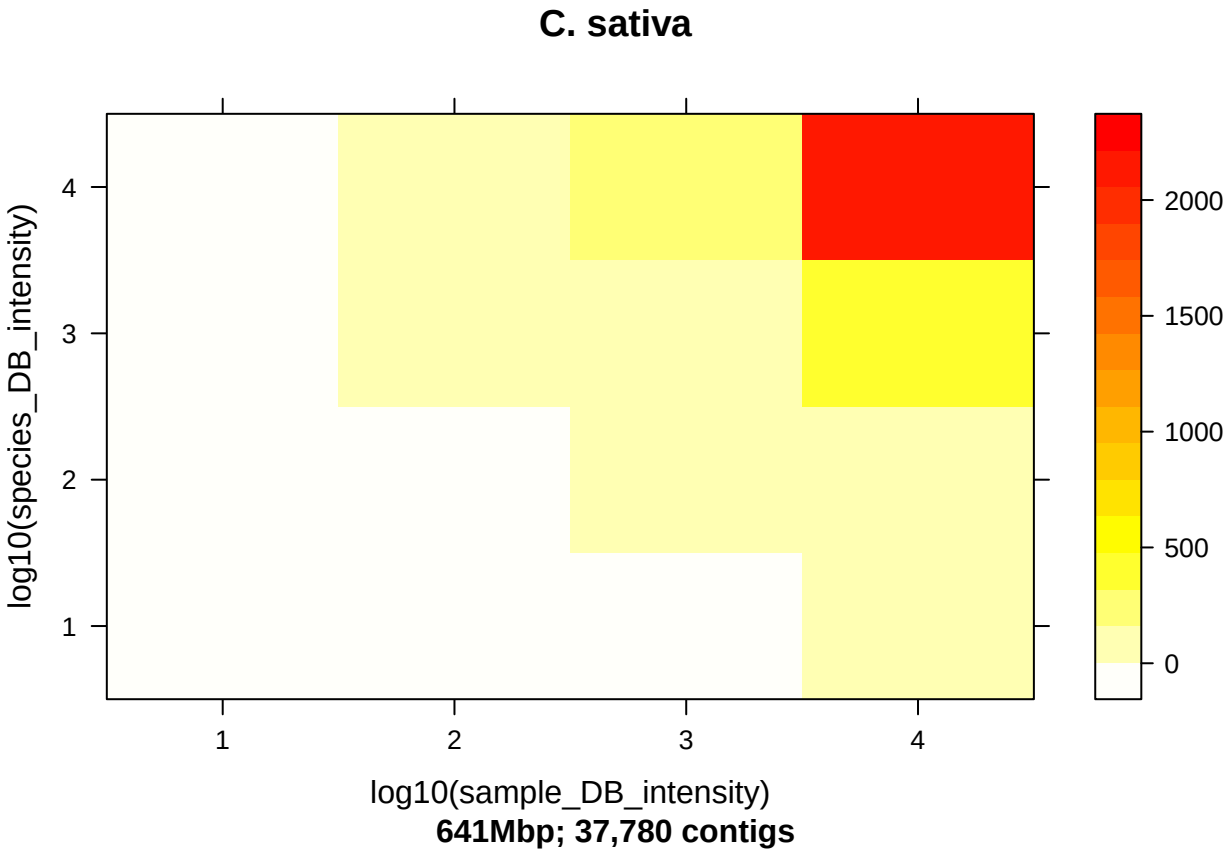

```
levelplot(total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$
```




```
levelplot(total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$
```

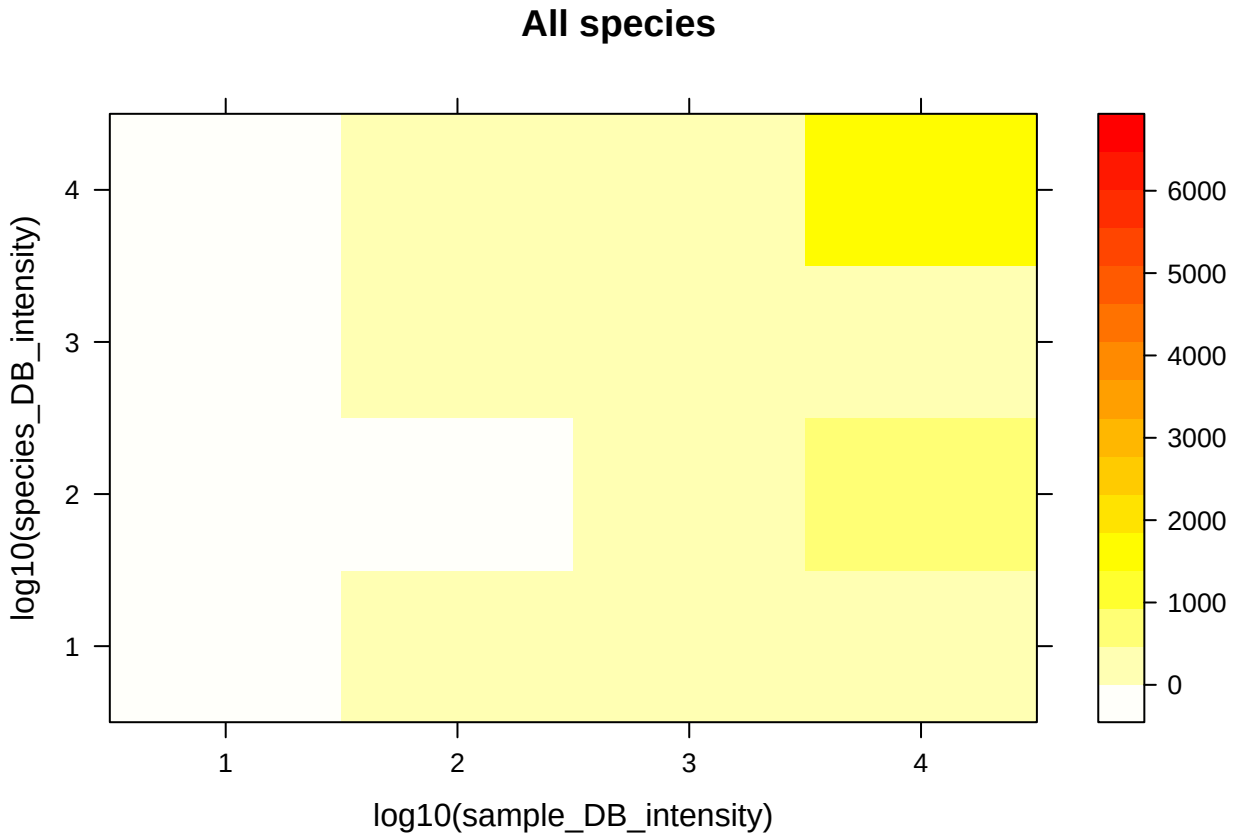



```
levelplot(total_hits ~ log10(sample_DB_intensity)*log10(species_DB_intensity),data=in_silico[in_silico$
```

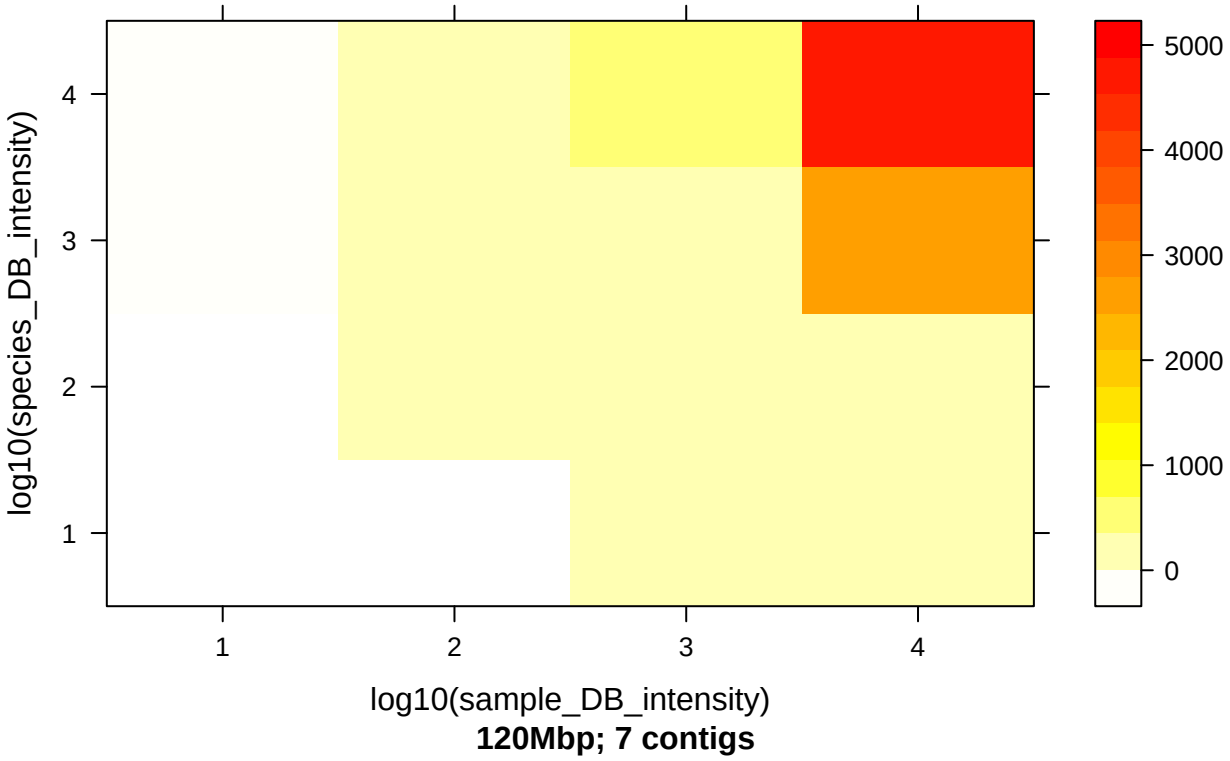


Okay, this is a little hard to interpret; now plot hit percentage (expectation) * total hits, e.g. number of expected positives:

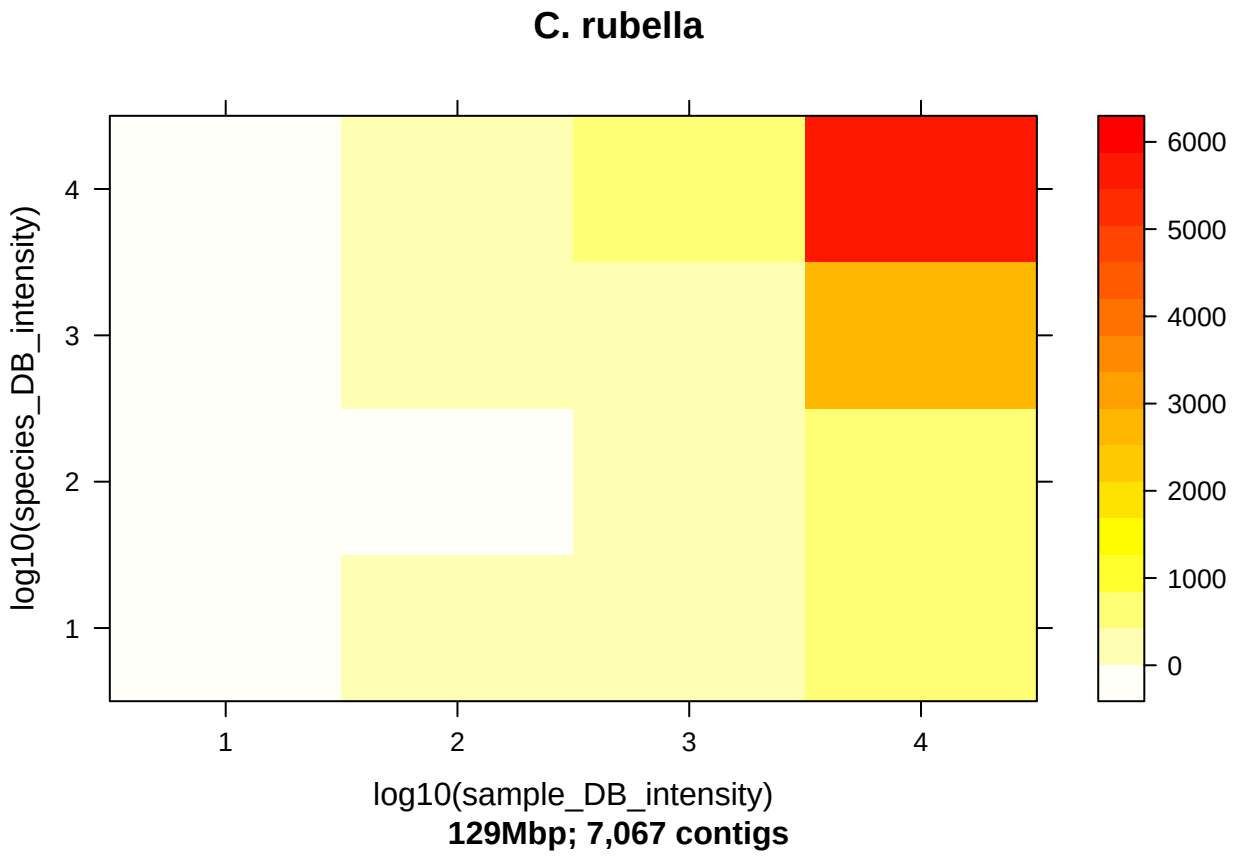
```
levelplot(total_hits * two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity)
```



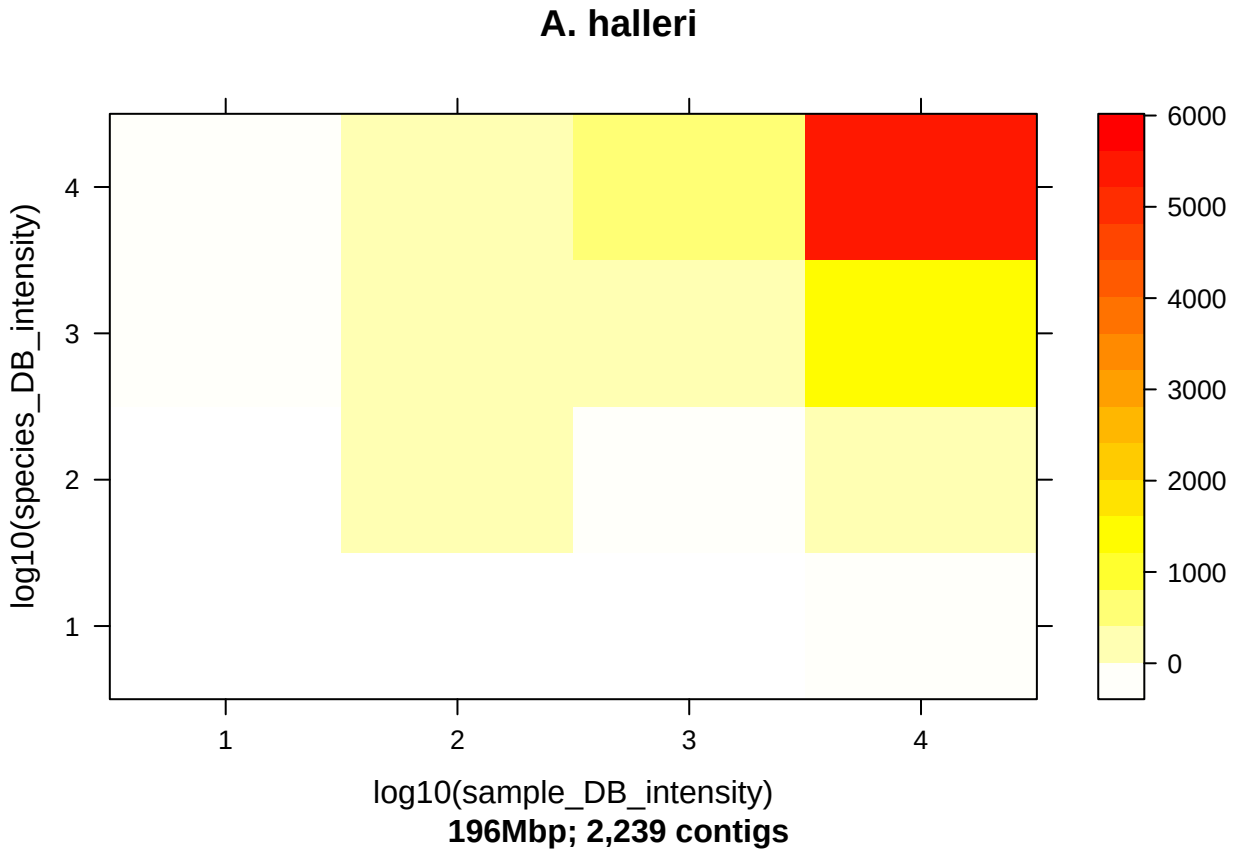

```
levelplot(total_hits * two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity)
```



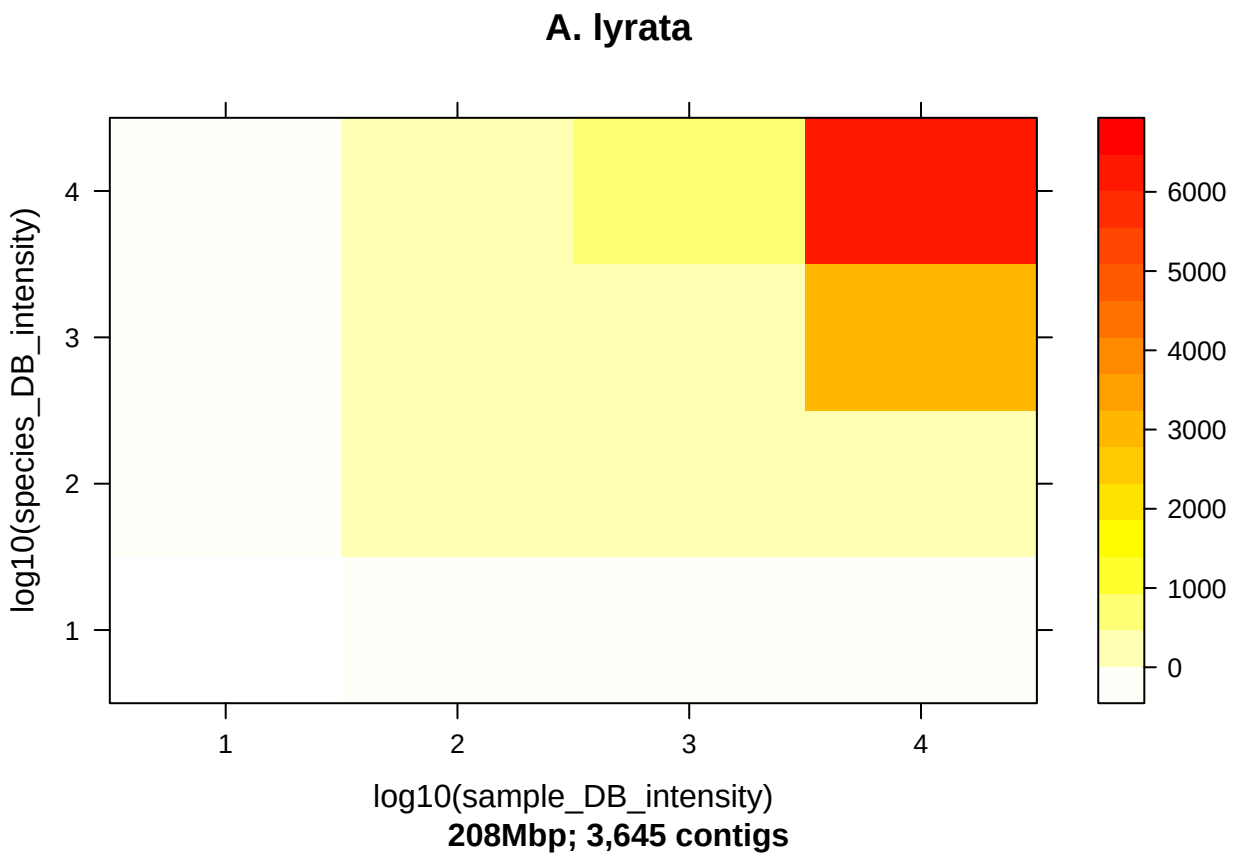

```
levelplot(total_hits * two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity)
```



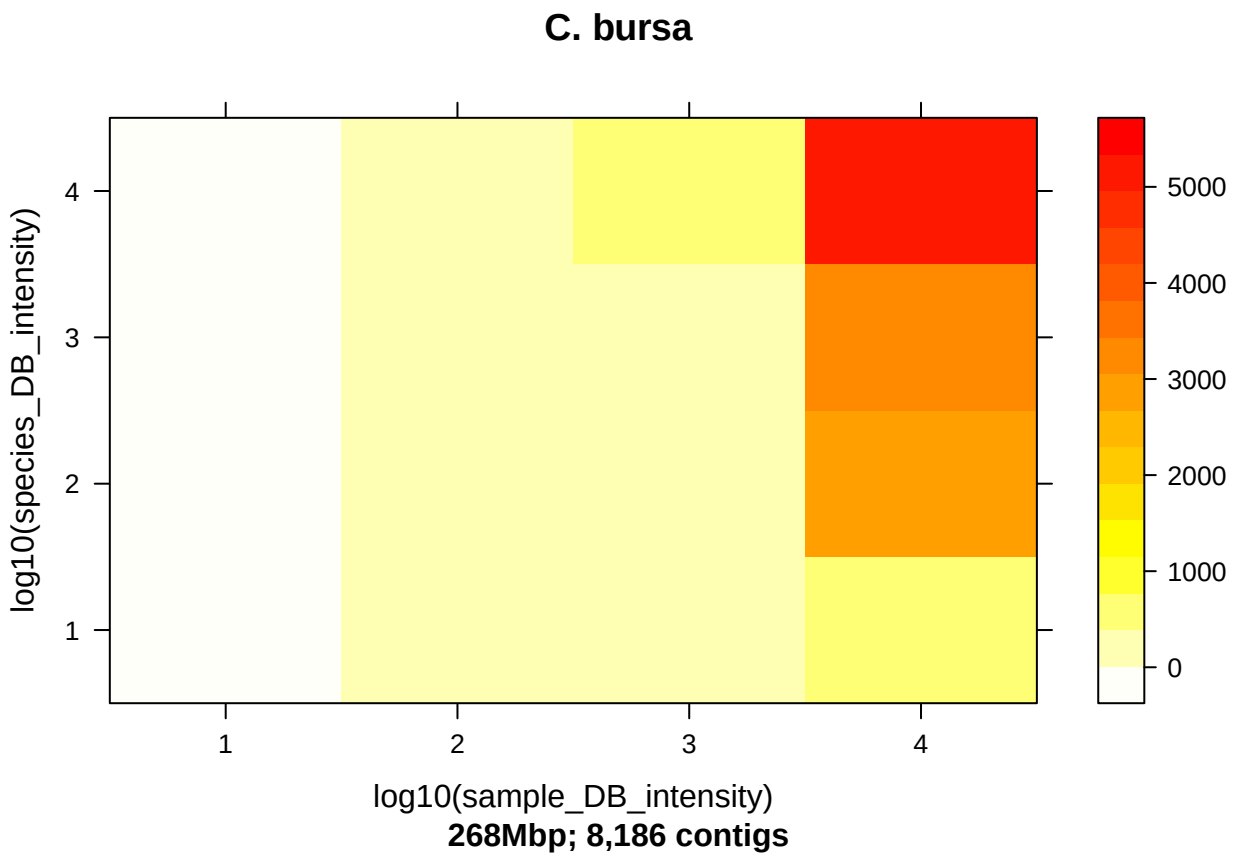

```
levelplot(total_hits * two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity)
```



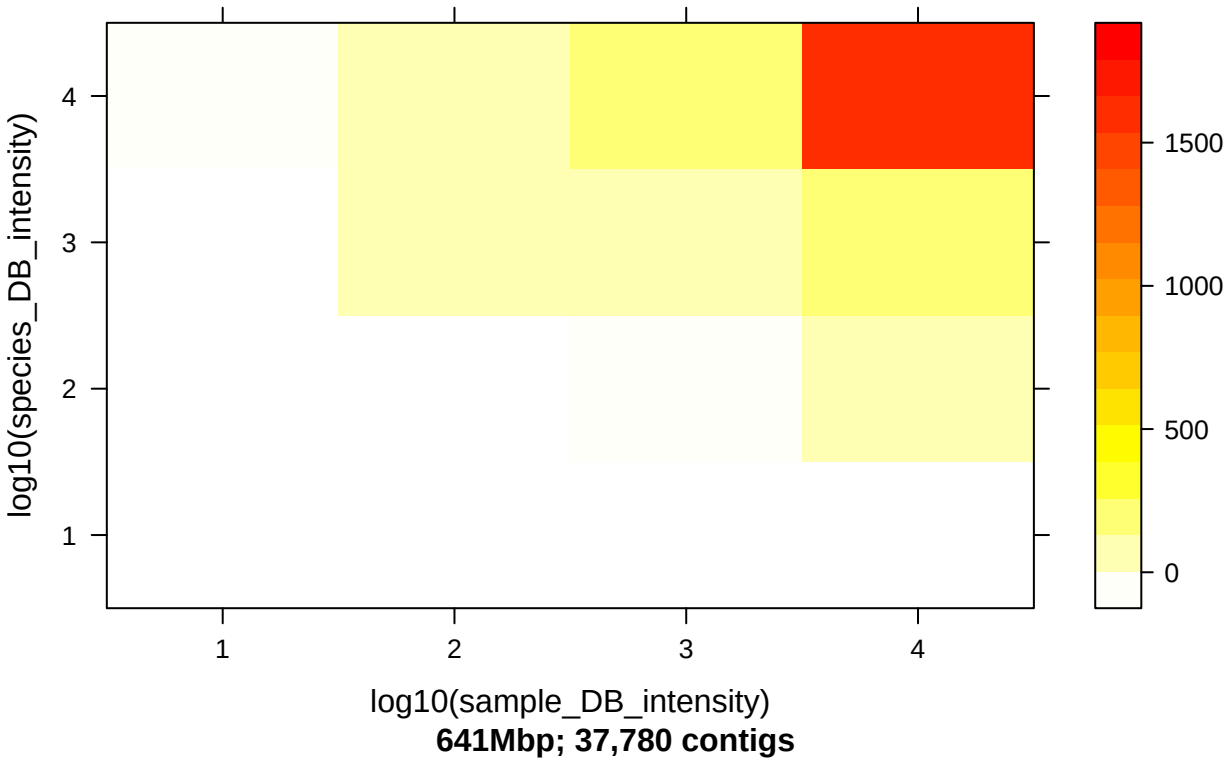

```
levelplot(total_hits * two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity)
```




```
levelplot(total_hits * two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity)
```




```
levelplot(total_hits * two_way_rate_with_cutoff ~ log10(sample_DB_intensity)*log10(species_DB_intensity)
```



Formally analyse the relationship with a GLM:

```
# 2-way with cutoff, by species too
model_total_positives=glm(total_hits * two_way_rate_with_cutoff ~ (sample_DB_intensity)*(species_DB_int
summary.glm(model_total_positives)
```

```
##
## Call:
## glm(formula = total_hits * two_way_rate_with_cutoff ~ (sample_DB_intensity) *
##     (species_DB_intensity) * TP_species, data = in_silico)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2715.43    -38.33     -1.55     16.03   1776.55
##
## Coefficients:
##                                                          Estimate
## (Intercept)                                             2.062e+00
## sample_DB_intensity                                     5.380e-02
## species_DB_intensity                                   -5.048e-04
## TP_speciesA-lyrata_04                                   2.757e+00
## TP_speciesA-thaliana_02                                 1.157e+01
## TP_speciesC-bursa_05                                   -1.546e+00
## TP_speciesC-rubella_03                                  7.988e-02
## TP_speciesC-sativa_06                                   4.931e-01
## sample_DB_intensity:species_DB_intensity                4.989e-05
## sample_DB_intensity:TP_speciesA-lyrata_04               8.872e-02
## sample_DB_intensity:TP_speciesA-thaliana_02             6.192e-02
```

```
## sample_DB_intensity:TP_speciesC-bursa_05                            2.174e-01
## sample_DB_intensity:TP_speciesC-rubella_03                          4.144e-02
## sample_DB_intensity:TP_speciesC-sativa_06                          -4.192e-02
## species_DB_intensity:TP_speciesA-lyrata_04                         -3.684e-05
## species_DB_intensity:TP_speciesA-thaliana_02                       -7.898e-04
## species_DB_intensity:TP_speciesC-bursa_05                           9.445e-04
## species_DB_intensity:TP_speciesC-rubella_03                         4.725e-04
## species_DB_intensity:TP_speciesC-sativa_06                         -6.095e-04
## sample_DB_intensity:species_DB_intensity:TP_speciesA-lyrata_04      5.694e-07
## sample_DB_intensity:species_DB_intensity:TP_speciesA-thaliana_02 -1.255e-05
## sample_DB_intensity:species_DB_intensity:TP_speciesC-bursa_05      -2.477e-05
## sample_DB_intensity:species_DB_intensity:TP_speciesC-rubella_03    -1.332e-06
## sample_DB_intensity:species_DB_intensity:TP_speciesC-sativa_06     -3.511e-05
##                                                                   Std. Error
## (Intercept)                                                        2.207e+01
## sample_DB_intensity                                                3.513e-03
## species_DB_intensity                                               3.210e-03
## TP_speciesA-lyrata_04                                              2.959e+01
## TP_speciesA-thaliana_02                                            3.299e+01
## TP_speciesC-bursa_05                                               2.659e+01
## TP_speciesC-rubella_03                                             2.668e+01
## TP_speciesC-sativa_06                                              6.272e+01
## sample_DB_intensity:species_DB_intensity                           5.746e-07
## sample_DB_intensity:TP_speciesA-lyrata_04                          4.836e-03
## sample_DB_intensity:TP_speciesA-thaliana_02                        5.131e-03
## sample_DB_intensity:TP_speciesC-bursa_05                           4.464e-03
## sample_DB_intensity:TP_speciesC-rubella_03                         4.482e-03
## sample_DB_intensity:TP_speciesC-sativa_06                          7.934e-03
## species_DB_intensity:TP_speciesA-lyrata_04                         4.426e-03
## species_DB_intensity:TP_speciesA-thaliana_02                       4.725e-03
## species_DB_intensity:TP_speciesC-bursa_05                          4.227e-03
## species_DB_intensity:TP_speciesC-rubella_03                        4.236e-03
## species_DB_intensity:TP_speciesC-sativa_06                         7.375e-03
## sample_DB_intensity:species_DB_intensity:TP_speciesA-lyrata_04     8.041e-07
## sample_DB_intensity:species_DB_intensity:TP_speciesA-thaliana_02   8.273e-07
## sample_DB_intensity:species_DB_intensity:TP_speciesC-bursa_05      7.802e-07
## sample_DB_intensity:species_DB_intensity:TP_speciesC-rubella_03    7.816e-07
## sample_DB_intensity:species_DB_intensity:TP_speciesC-sativa_06     1.051e-06
##                                                                   t value
## (Intercept)                                                         0.093
## sample_DB_intensity                                                15.312
## species_DB_intensity                                               -0.157
## TP_speciesA-lyrata_04                                               0.093
## TP_speciesA-thaliana_02                                             0.351
## TP_speciesC-bursa_05                                               -0.058
## TP_speciesC-rubella_03                                              0.003
## TP_speciesC-sativa_06                                               0.008
## sample_DB_intensity:species_DB_intensity                           86.827
## sample_DB_intensity:TP_speciesA-lyrata_04                          18.347
## sample_DB_intensity:TP_speciesA-thaliana_02                        12.067
## sample_DB_intensity:TP_speciesC-bursa_05                           48.711
## sample_DB_intensity:TP_speciesC-rubella_03                          9.245
## sample_DB_intensity:TP_speciesC-sativa_06                          -5.283
## species_DB_intensity:TP_speciesA-lyrata_04                         -0.008
```

```
## species_DB_intensity:TP_speciesA-thaliana_02                      -0.167
## species_DB_intensity:TP_speciesC-bursa_05                          0.223
## species_DB_intensity:TP_speciesC-rubella_03                        0.112
## species_DB_intensity:TP_speciesC-sativa_06                        -0.083
## sample_DB_intensity:species_DB_intensity:TP_speciesA-lyrata_04     0.708
## sample_DB_intensity:species_DB_intensity:TP_speciesA-thaliana_02 -15.171
## sample_DB_intensity:species_DB_intensity:TP_speciesC-bursa_05    -31.748
## sample_DB_intensity:species_DB_intensity:TP_speciesC-rubella_03   -1.705
## sample_DB_intensity:species_DB_intensity:TP_speciesC-sativa_06   -33.402
##                                                                  Pr(>|t|)
## (Intercept)                                                        0.9256
## sample_DB_intensity                                              < 2e-16
## species_DB_intensity                                               0.8750
## TP_speciesA-lyrata_04                                              0.9258
## TP_speciesA-thaliana_02                                            0.7259
## TP_speciesC-bursa_05                                               0.9536
## TP_speciesC-rubella_03                                             0.9976
## TP_speciesC-sativa_06                                              0.9937
## sample_DB_intensity:species_DB_intensity                         < 2e-16
## sample_DB_intensity:TP_speciesA-lyrata_04                        < 2e-16
## sample_DB_intensity:TP_speciesA-thaliana_02                      < 2e-16
## sample_DB_intensity:TP_speciesC-bursa_05                         < 2e-16
## sample_DB_intensity:TP_speciesC-rubella_03                       < 2e-16
## sample_DB_intensity:TP_speciesC-sativa_06                       1.31e-07
## species_DB_intensity:TP_speciesA-lyrata_04                         0.9934
## species_DB_intensity:TP_speciesA-thaliana_02                       0.8672
## species_DB_intensity:TP_speciesC-bursa_05                          0.8232
## species_DB_intensity:TP_speciesC-rubella_03                        0.9112
## species_DB_intensity:TP_speciesC-sativa_06                         0.9341
## sample_DB_intensity:species_DB_intensity:TP_speciesA-lyrata_04     0.4789
## sample_DB_intensity:species_DB_intensity:TP_speciesA-thaliana_02  < 2e-16
## sample_DB_intensity:species_DB_intensity:TP_speciesC-bursa_05     < 2e-16
## sample_DB_intensity:species_DB_intensity:TP_speciesC-rubella_03    0.0883
## sample_DB_intensity:species_DB_intensity:TP_speciesC-sativa_06    < 2e-16
##
## (Intercept)
## sample_DB_intensity                                              ***
## species_DB_intensity
## TP_speciesA-lyrata_04
## TP_speciesA-thaliana_02
## TP_speciesC-bursa_05
## TP_speciesC-rubella_03
## TP_speciesC-sativa_06
## sample_DB_intensity:species_DB_intensity                         ***
## sample_DB_intensity:TP_speciesA-lyrata_04                        ***
## sample_DB_intensity:TP_speciesA-thaliana_02                      ***
## sample_DB_intensity:TP_speciesC-bursa_05                         ***
## sample_DB_intensity:TP_speciesC-rubella_03                       ***
## sample_DB_intensity:TP_speciesC-sativa_06                        ***
## species_DB_intensity:TP_speciesA-lyrata_04
## species_DB_intensity:TP_speciesA-thaliana_02
## species_DB_intensity:TP_speciesC-bursa_05
## species_DB_intensity:TP_speciesC-rubella_03
## species_DB_intensity:TP_speciesC-sativa_06
```

```
## sample_DB_intensity:species_DB_intensity:TP_speciesA-lyrata_04
## sample_DB_intensity:species_DB_intensity:TP_speciesA-thaliana_02 ***
## sample_DB_intensity:species_DB_intensity:TP_speciesC-bursa_05    ***
## sample_DB_intensity:species_DB_intensity:TP_speciesC-rubella_03  .
## sample_DB_intensity:species_DB_intensity:TP_speciesC-sativa_06   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 140508)
##
##     Null deviance: 1.7215e+10  on 6551  degrees of freedom
## Residual deviance: 9.1724e+08  on 6528  degrees of freedom
##   (3816 observations deleted due to missingness)
## AIC: 96281
##
## Number of Fisher Scoring iterations: 2
```

```
summary(aov(model_total_positives))
```

```
##                                                    Df    Sum Sq
## sample_DB_intensity                                 1 8.456e+09
## species_DB_intensity                                1 1.877e+09
## TP_species                                          5 1.027e+09
## sample_DB_intensity:species_DB_intensity            1 3.311e+09
## sample_DB_intensity:TP_species                      5 9.548e+08
## species_DB_intensity:TP_species                     5 3.147e+08
## sample_DB_intensity:species_DB_intensity:TP_species    5 3.567e+08
## Residuals                                        6528 9.172e+08
##                                                     Mean Sq F value
## sample_DB_intensity                              8.456e+09 60180.3
## species_DB_intensity                             1.877e+09 13360.8
## TP_species                                       2.055e+08  1462.4
## sample_DB_intensity:species_DB_intensity         3.311e+09 23561.1
## sample_DB_intensity:TP_species                   1.910e+08  1359.1
## species_DB_intensity:TP_species                  6.294e+07   448.0
## sample_DB_intensity:species_DB_intensity:TP_species 7.134e+07   507.7
## Residuals                                        1.405e+05
##                                                  Pr(>F)
## sample_DB_intensity                              <2e-16 ***
## species_DB_intensity                             <2e-16 ***
## TP_species                                       <2e-16 ***
## sample_DB_intensity:species_DB_intensity         <2e-16 ***
## sample_DB_intensity:TP_species                   <2e-16 ***
## species_DB_intensity:TP_species                  <2e-16 ***
## sample_DB_intensity:species_DB_intensity:TP_species <2e-16 ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3816 observations deleted due to missingness
```

```
# two-way rate with cutoff, no species
model_total_positives_nosp=glm(total_hits * two_way_rate_with_cutoff ~ (sample_DB_intensity)*(species_DB
summary(aov(model_total_positives_nosp))
```

```
##                                        Df    Sum Sq   Mean Sq F value
```

```
## sample_DB_intensity                          1 8.456e+09 8.456e+09   15823
## species_DB_intensity                         1 1.877e+09 1.877e+09    3513
## sample_DB_intensity:species_DB_intensity     1 3.382e+09 3.382e+09    6329
## Residuals                                 6548 3.499e+09 5.344e+05
##                                                Pr(>F)
## sample_DB_intensity                          <2e-16 ***
## species_DB_intensity                         <2e-16 ***
## sample_DB_intensity:species_DB_intensity     <2e-16 ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3816 observations deleted due to missingness
```

```
summary(model_total_positives_nosp)
```

```
##
## Call:
## glm(formula = total_hits * two_way_rate_with_cutoff ~ (sample_DB_intensity) *
##     (species_DB_intensity), data = in_silico)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3445.4    -50.2     -0.6     83.2   2185.1
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                8.777e+00  1.563e+01   0.562
## sample_DB_intensity                        1.312e-01  2.644e-03  49.615
## species_DB_intensity                      -8.070e-04  2.478e-03  -0.326
## sample_DB_intensity:species_DB_intensity   3.592e-05  4.515e-07  79.554
##                                             Pr(>|t|)
## (Intercept)                                    0.574
## sample_DB_intensity                          <2e-16 ***
## species_DB_intensity                           0.745
## sample_DB_intensity:species_DB_intensity     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 534399.9)
##
##     Null deviance: 1.7215e+10  on 6551  degrees of freedom
## Residual deviance: 3.4993e+09  on 6548  degrees of freedom
##   (3816 observations deleted due to missingness)
## AIC: 105013
##
## Number of Fisher Scoring iterations: 2
```

```
# 2-way, by species too
model_total_positives=glm(total_hits * two_way_rate ~ (sample_DB_intensity)*(species_DB_intensity)*TP_sp
summary(aov(model_total_positives))
```

```
##                                                  Df    Sum Sq
## sample_DB_intensity                               1 8.312e+09
## species_DB_intensity                              1 1.752e+09
## TP_species                                        5 9.515e+08
```

```
## sample_DB_intensity:species_DB_intensity                1 3.241e+09
## sample_DB_intensity:TP_species                          5 9.896e+08
## species_DB_intensity:TP_species                         5 2.323e+08
## sample_DB_intensity:species_DB_intensity:TP_species     5 2.633e+08
## Residuals                                            6798 7.738e+08
##                                                        Mean Sq F value
## sample_DB_intensity                                 8.312e+09 73024.4
## species_DB_intensity                                1.752e+09 15387.5
## TP_species                                          1.903e+08  1671.7
## sample_DB_intensity:species_DB_intensity            3.241e+09 28474.6
## sample_DB_intensity:TP_species                      1.979e+08  1738.7
## species_DB_intensity:TP_species                     4.646e+07   408.2
## sample_DB_intensity:species_DB_intensity:TP_species 5.265e+07   462.5
## Residuals                                           1.138e+05
##                                                     Pr(>F)
## sample_DB_intensity                                 <2e-16 ***
## species_DB_intensity                                <2e-16 ***
## TP_species                                          <2e-16 ***
## sample_DB_intensity:species_DB_intensity            <2e-16 ***
## sample_DB_intensity:TP_species                      <2e-16 ***
## species_DB_intensity:TP_species                     <2e-16 ***
## sample_DB_intensity:species_DB_intensity:TP_species <2e-16 ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3546 observations deleted due to missingness
```

```
# two-way rate with cutoff, no species
model_total_positives_nosp=glm(total_hits * two_way_rate ~ (sample_DB_intensity)*(species_DB_intensity)
summary(aov(model_total_positives_nosp))
```

```
##                                          Df    Sum Sq   Mean Sq F value
## sample_DB_intensity                       1 8.312e+09 8.312e+09   18138
## species_DB_intensity                      1 1.752e+09 1.752e+09    3822
## sample_DB_intensity:species_DB_intensity  1 3.327e+09 3.327e+09    7260
## Residuals                              6818 3.125e+09 4.583e+05
##                                          Pr(>F)
## sample_DB_intensity                      <2e-16 ***
## species_DB_intensity                     <2e-16 ***
## sample_DB_intensity:species_DB_intensity <2e-16 ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3546 observations deleted due to missingness
```

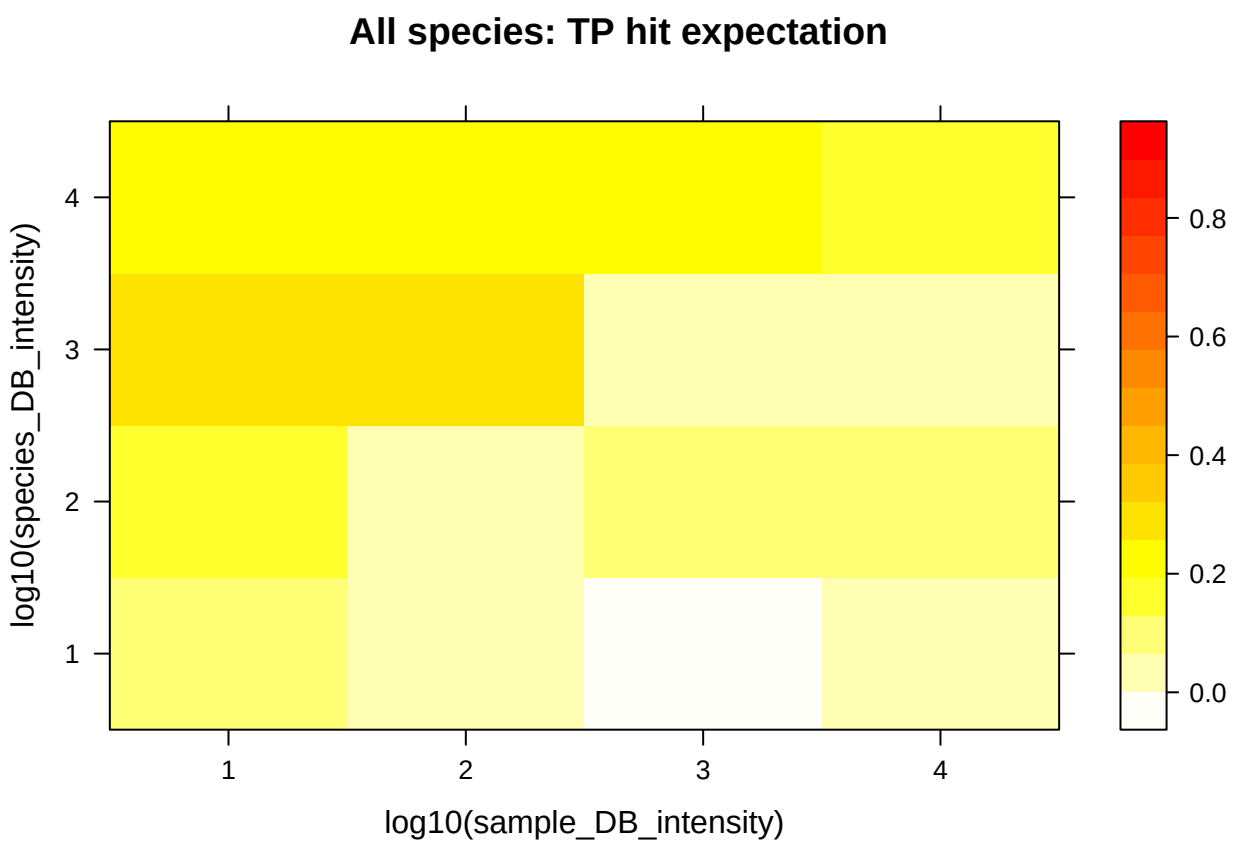
```
summary(model_total_positives_nosp)
```

```
##
## Call:
## glm(formula = total_hits * two_way_rate ~ (sample_DB_intensity) *
##     (species_DB_intensity), data = in_silico)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3429.9    -37.6     -0.9     66.3   2141.6
```

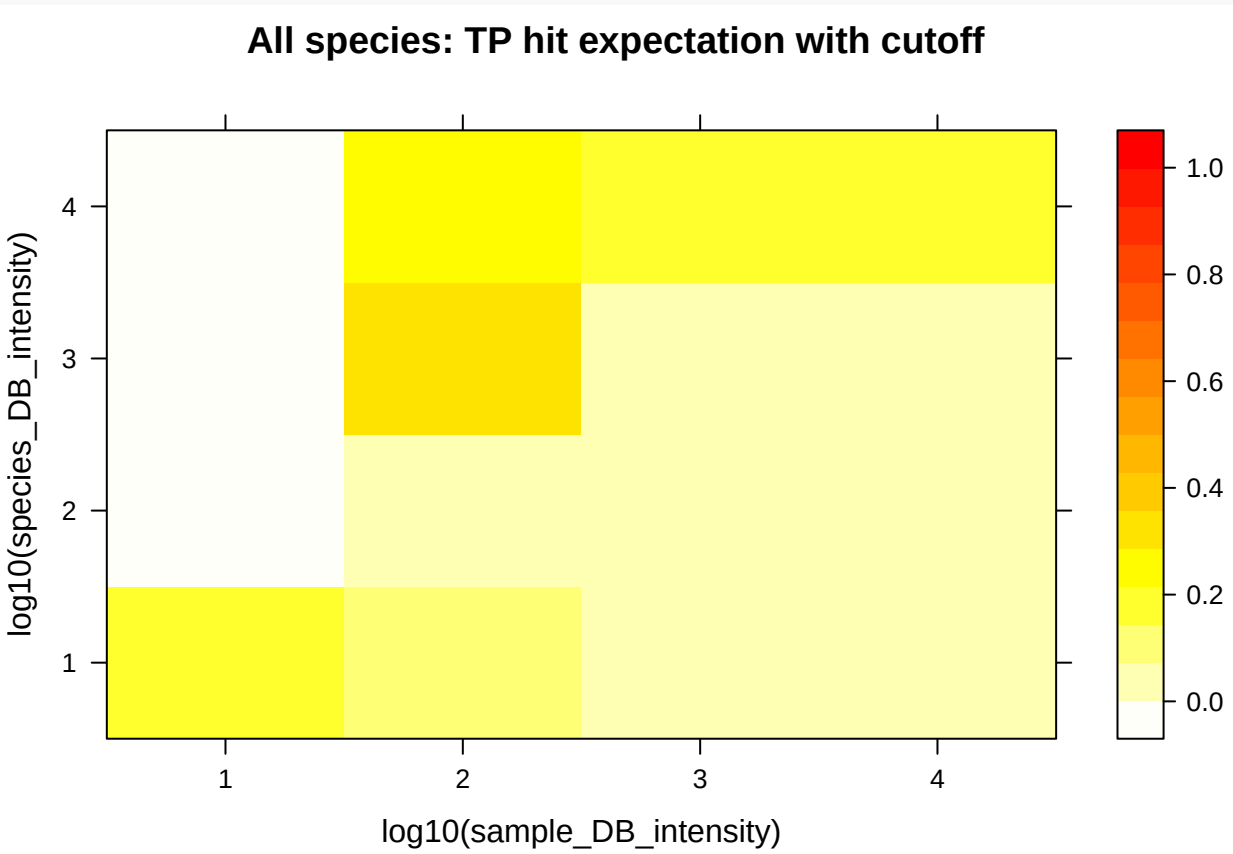
```
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                             7.192e+00  1.394e+01   0.516
## sample_DB_intensity                     1.286e-01  2.410e-03  53.358
## species_DB_intensity                   -7.129e-04  2.229e-03  -0.320
## sample_DB_intensity:species_DB_intensity  3.525e-05  4.137e-07  85.203
##                                        Pr(>|t|)
## (Intercept)                               0.606
## sample_DB_intensity                      <2e-16 ***
## species_DB_intensity                      0.749
## sample_DB_intensity:species_DB_intensity  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 458292.1)
##
##     Null deviance: 1.6516e+10  on 6821  degrees of freedom
## Residual deviance: 3.1246e+09  on 6818  degrees of freedom
##   (3546 observations deleted due to missingness)
## AIC: 108293
##
## Number of Fisher Scoring iterations: 2
```

And what we might really want to know is the expectation that a read in a sequencing run of length n will be a positive:

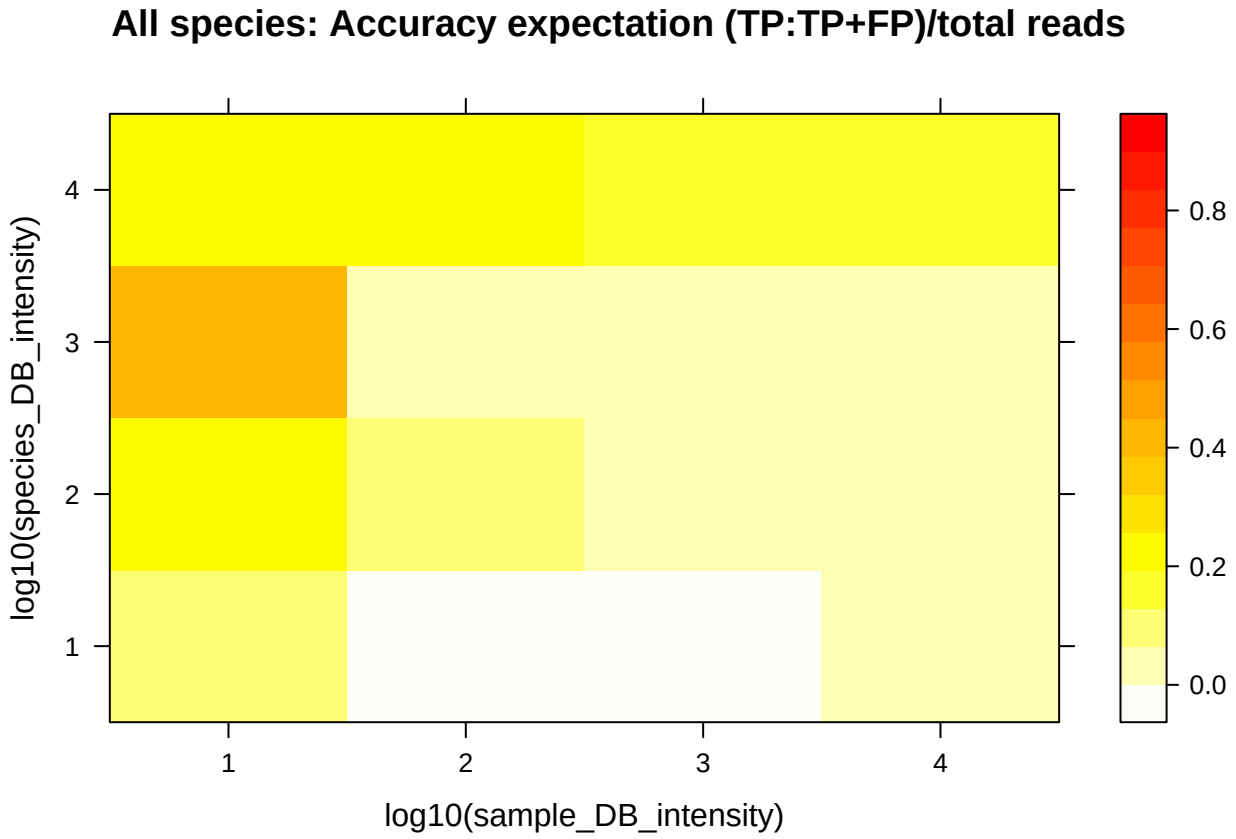
```
levelplot((total_hits * two_way_rate)/sample_DB_intensity ~ log10(sample_DB_intensity)*log10(species_DB_
```



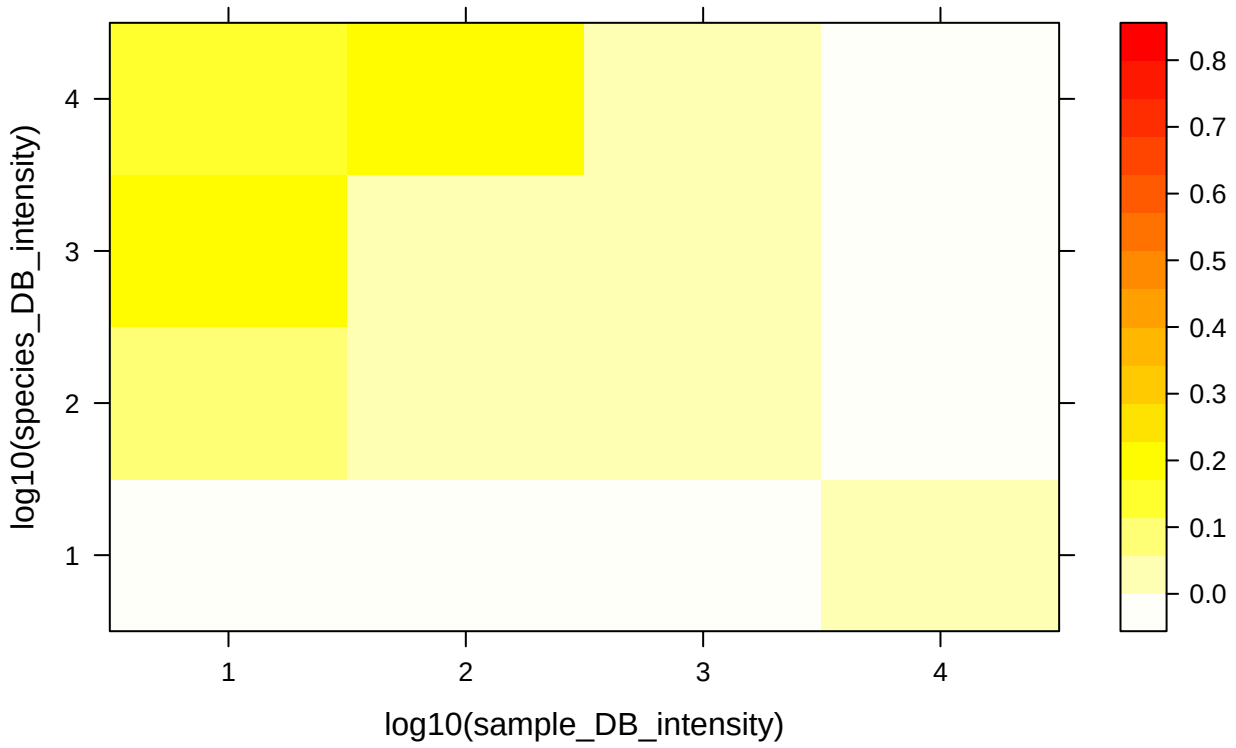
```
levelplot((total_hits * two_way_rate_with_cutoff)/sample_DB_intensity ~ log10(sample_DB_intensity)*log1
```



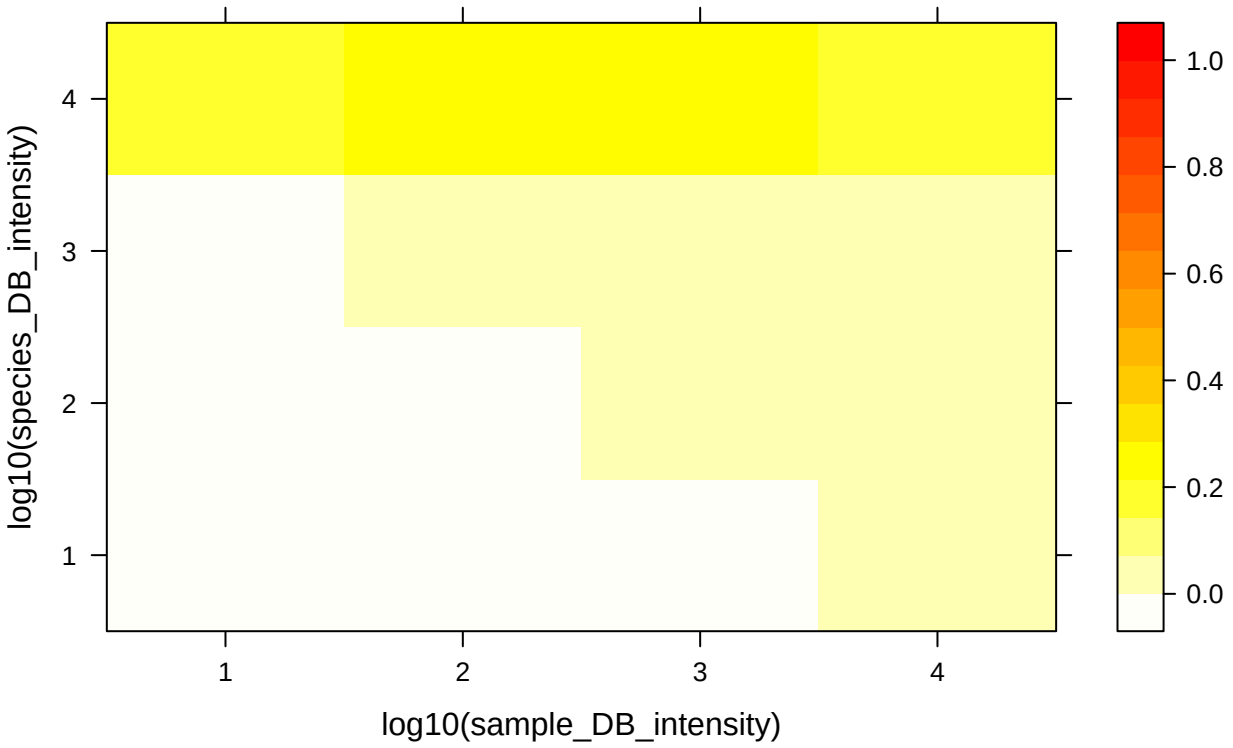

```
levelplot(((total_hits * ((one_way_TP + two_way_TP)/(one_way_FP + two_way_FP+one_way_TP+two_way_TP)))/sa
```




```
levelplot((( ((one_way_TP + two_way_TP)/(one_way_FP + two_way_FP)))/sample_DB_intensity) ~ log10(sample_
```

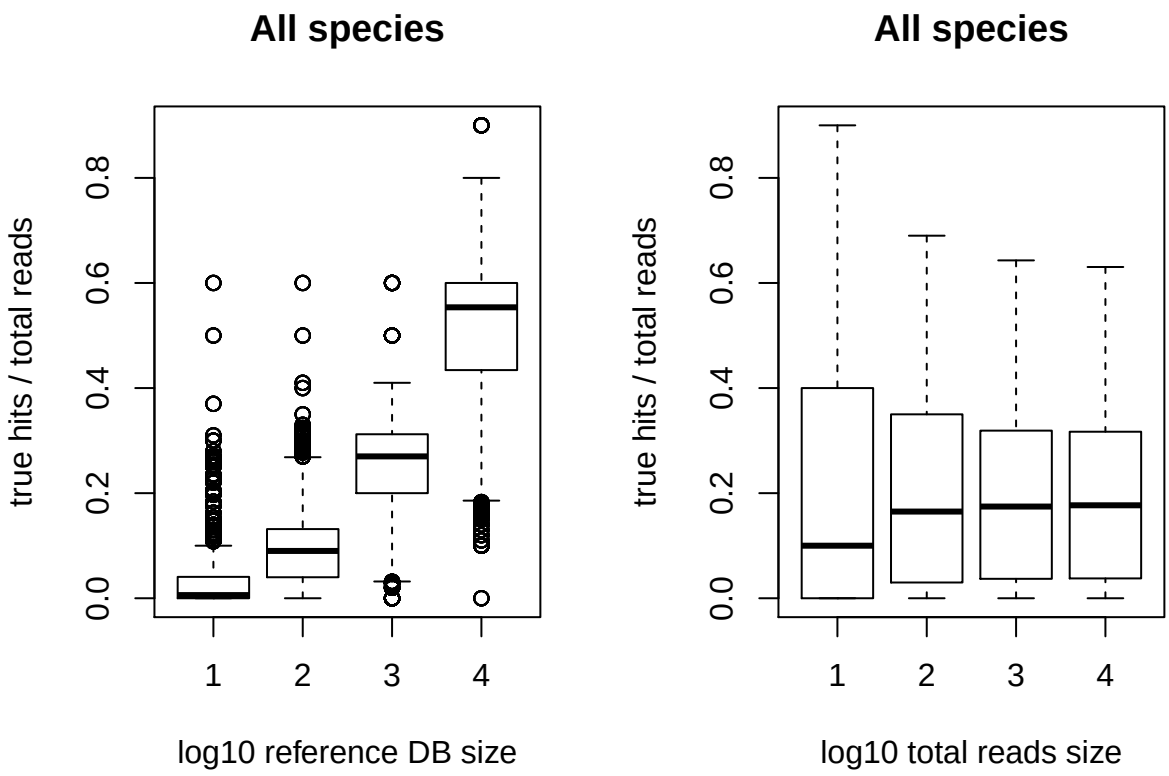



```
levelplot((total_hits /sample_DB_intensity) ~ log10(sample_DB_intensity)*log10(species_DB_intensity),da
```



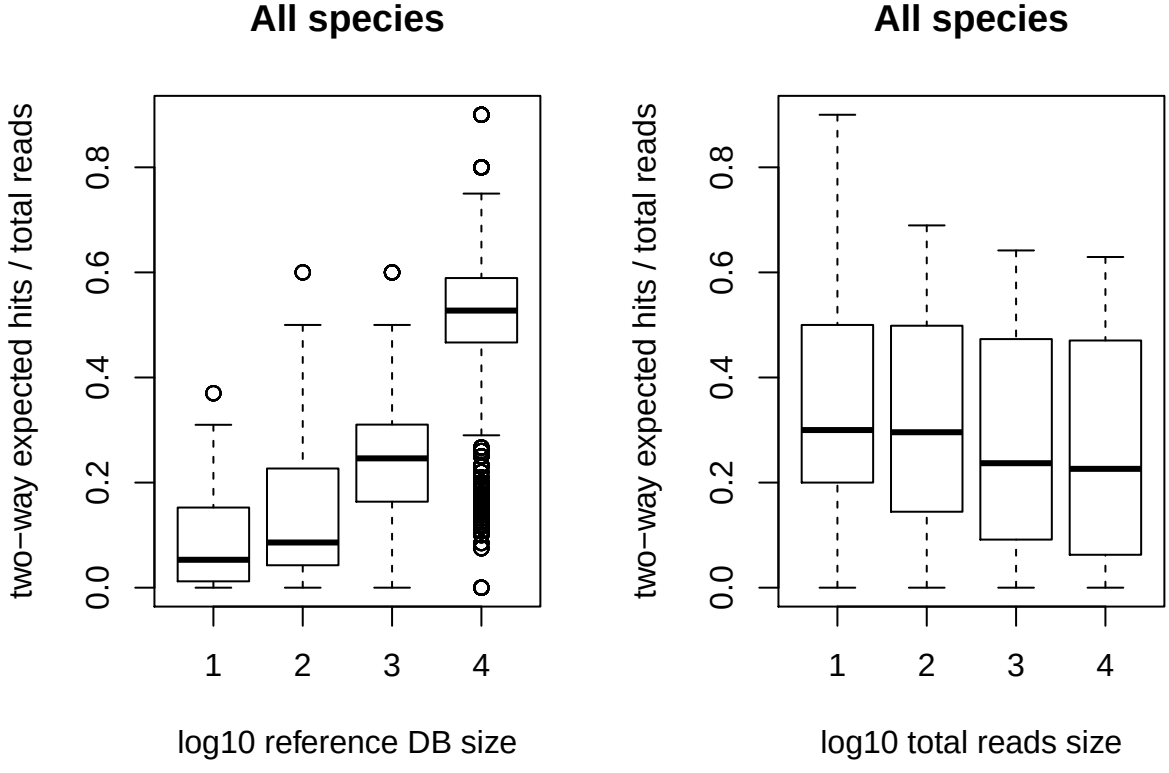
Let's break down the expected number of good hits a bit more simply:

```
par(mfrow=c(1,2))
boxplot((( ((one_way_TP + two_way_TP)))/sample_DB_intensity) ~ log10(species_DB_intensity),data=in_silic
boxplot((( ((one_way_TP + two_way_TP)))/sample_DB_intensity) ~ log10(sample_DB_intensity),data=in_silic
```



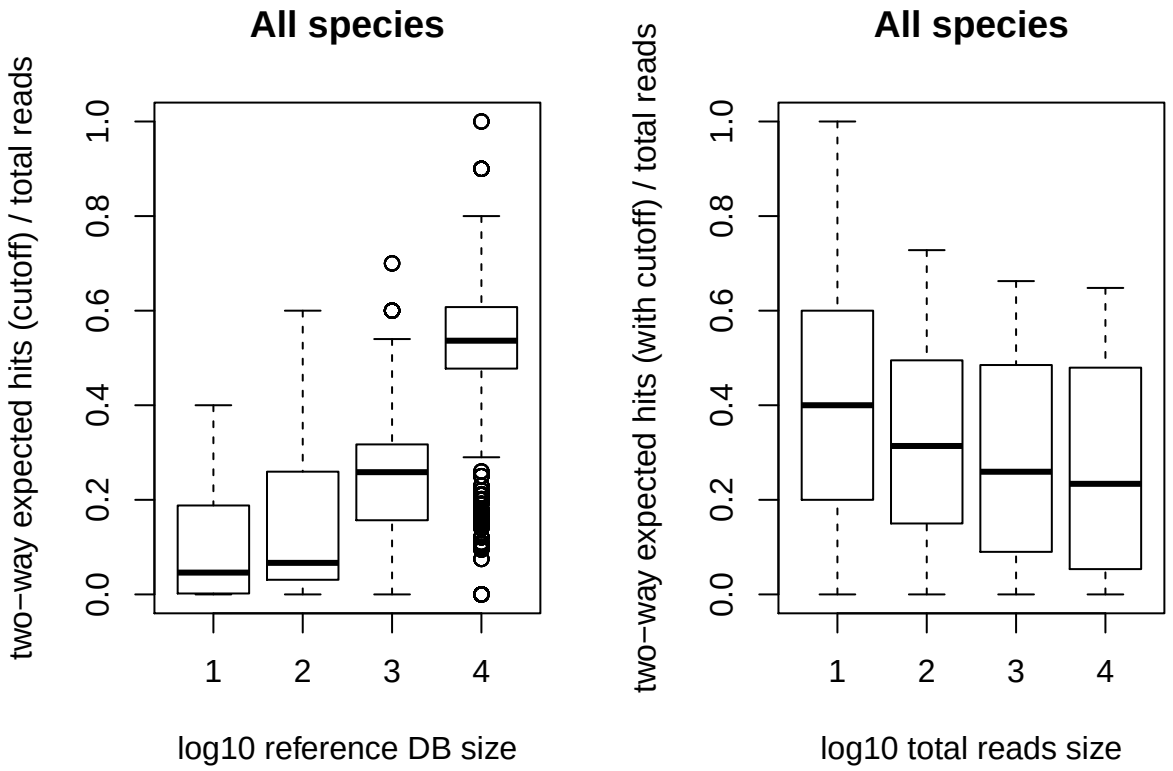
As above but using two-way hit rate:

```
par(mfrow=c(1,2))
boxplot((total_hits * two_way_rate)/sample_DB_intensity ~ log10(species_DB_intensity),data=in_silico, c
boxplot((total_hits * two_way_rate)/sample_DB_intensity ~ log10(sample_DB_intensity),data=in_silico, co
```



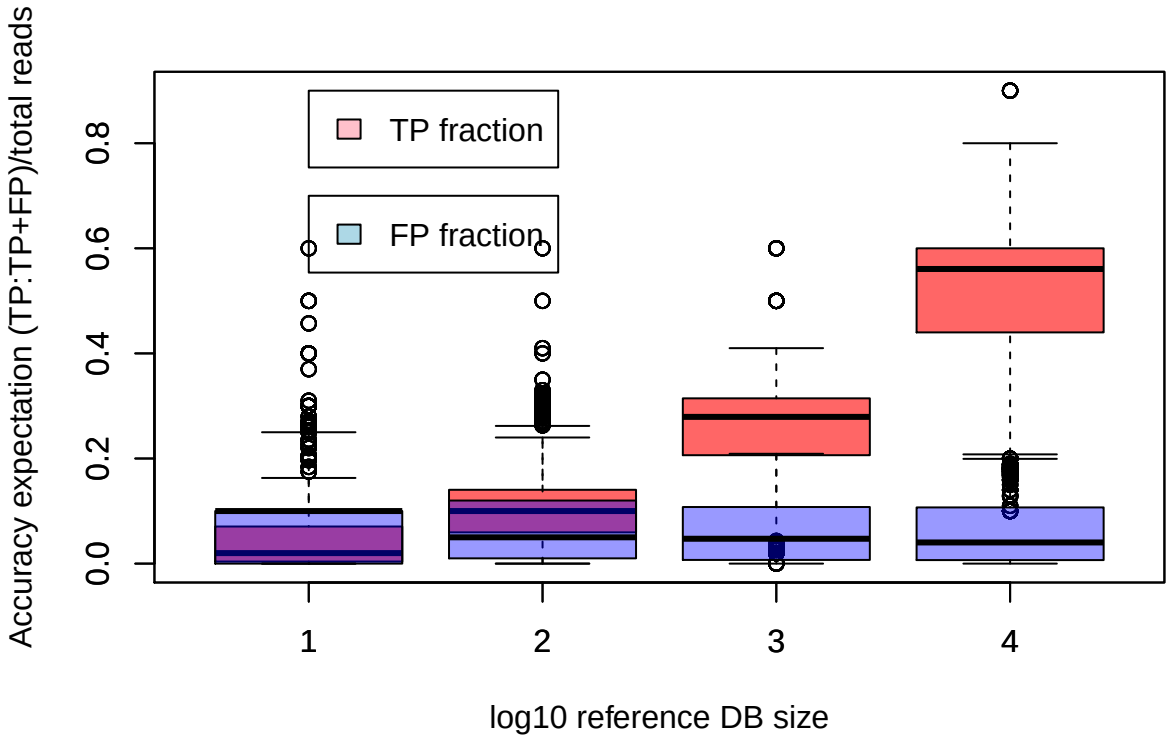
As above but using two-way hit rate, with cutoff > 50:

```
par(mfrow=c(1,2))
boxplot((total_hits * two_way_rate_with_cutoff)/sample_DB_intensity ~ log10(species_DB_intensity),data=
boxplot((total_hits * two_way_rate_with_cutoff)/sample_DB_intensity ~ log10(sample_DB_intensity),data=i
```



As above but accuracy e.g. [(TP:TP+FP) / total reads]. This is probably closest to the headline 'effective ID rate', e.g. expectation that a given sequenced read produces a BLAST hit which is accurately a true positive.

```
#par(mfrow=c(1,2))
# TP rate in red
boxplot(((total_hits * ((one_way_TP + two_way_TP)/(one_way_FP + two_way_FP+one_way_TP+two_way_TP)))/samp
# FP rate in blue
boxplot(((total_hits * ((one_way_FP + two_way_FP)/(one_way_FP + two_way_FP+one_way_TP+two_way_TP)))/samp
legend(1,0.9,'TP fraction',fill='pink')
legend(1,0.7,'FP fraction',fill='light blue')
```



Of course we should ask the reciprocal question e.g. [(FP:TP+FP) / total reads], expectation that a given sequenced read produces a BLAST hit which is inaccurately a false positive.

```
par(mfrow=c(1,2))
boxplot(((total_hits * ((one_way_FP + two_way_FP)/(one_way_FP + two_way_FP+one_way_TP+two_way_TP)))/samp
boxplot(((total_hits * ((one_way_FP + two_way_FP)/(one_way_FP + two_way_FP+one_way_TP+two_way_TP)))/samp
```