Details about the linear model

Friederike Duendar

8/24/2017

They don't need to type with me, they should just follow along.

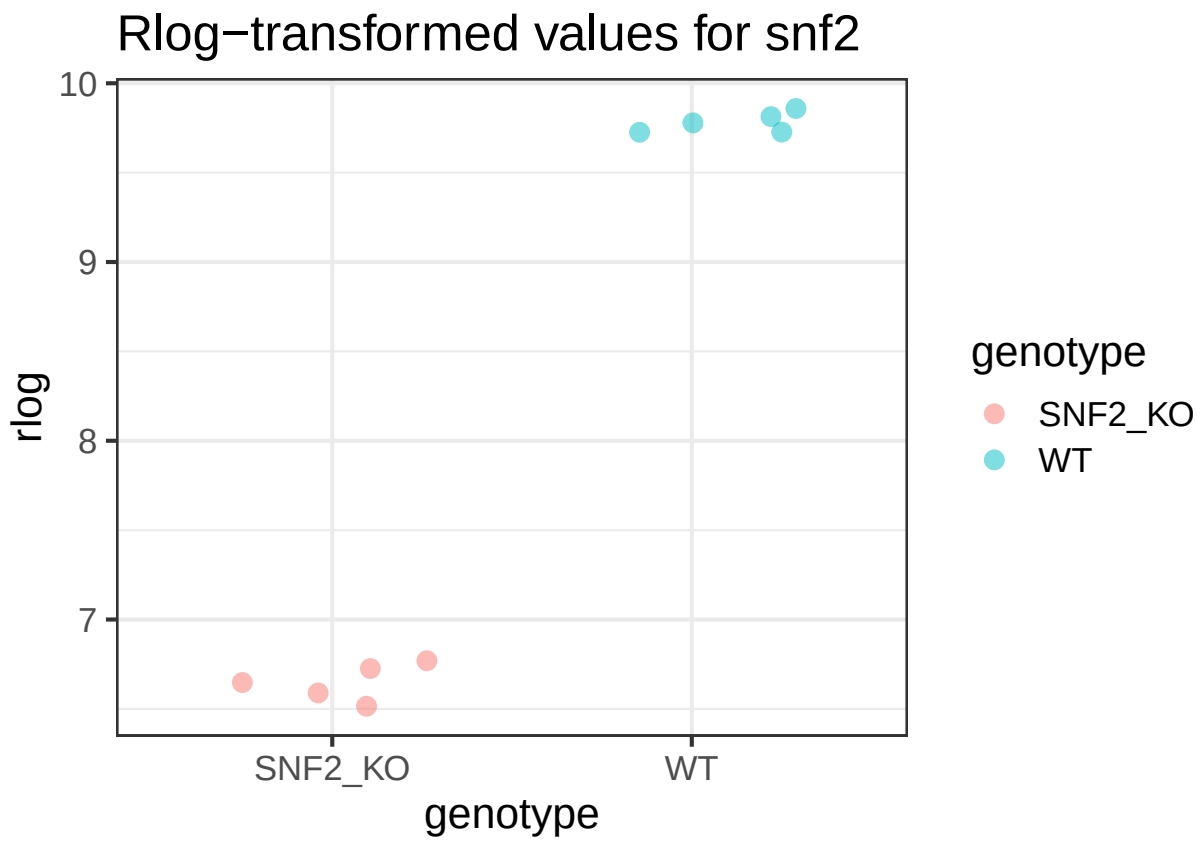
Goal: show what the testing is about using one example gene (snf2)

- extract rlog values for snf2 (YOR290C)

```
snf2 <- data.frame( rlog = assay(DESeq.rlog)["YOR290C",],
            sample = names( assay(DESeq.rlog)["YOR290C",] )
            )
snf2$genotype <- gsub("_[0-9]+", "", snf2$sample)
snf2$genotype <- ifelse(snf2$genotype == "SNF2", "SNF2_KO",  snf2$genotype)
```

```
library(ggplot2)
theme_set( theme_bw(base_size = 16) )

ggplot(snf2, aes(x = genotype, y = rlog, color = genotype)) +
  geom_jitter(width = .3, size = 3, alpha =.5) +
  ggtitle("Rlog-transformed values for snf2")
```



- in order to test whether the difference in expression is significant, we can do a t-test

```
t.test(rlog ~ genotype, data=snf2)
```

```
##
##  Welch Two Sample t-test
##
## data:  rlog by genotype
## t = -59.697, df = 6.2691, p-value = 7.067e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.258184 -3.004147
## sample estimates:
## mean in group SNF2_KO      mean in group WT
##              6.649520              9.780685
```

t-test tests the difference between the sample means

This would be equivalent to using a linear model of the following form:

```
lm(rlog ~ genotype, data = snf2)
```

```
##
## Call:
## lm(formula = rlog ~ genotype, data = snf2)
##
## Coefficients:
## (Intercept)   genotypeWT
##       6.650        3.131
```

Notice how the intercept equals the mean of group SNF2 in the t.test.

```
# let's assign the lm output to an object because there's lots of stuff in there
modelling_snf2 <- lm(rlog ~ genotype, data = snf2)
coef(modelling_snf2)
```

```
## (Intercept)  genotypeWT
##    6.649520    3.131165
```

The linear model is based on this function:

$Y = b0 + b1X + e$

where:

- Y corresponds to the rlog values
- X corresponds to the genotype (here: either SNF2 or WT)
- $b0$ is the intercept
- $b1$ is the term we're actually interested in, i.e. the difference between the two groups

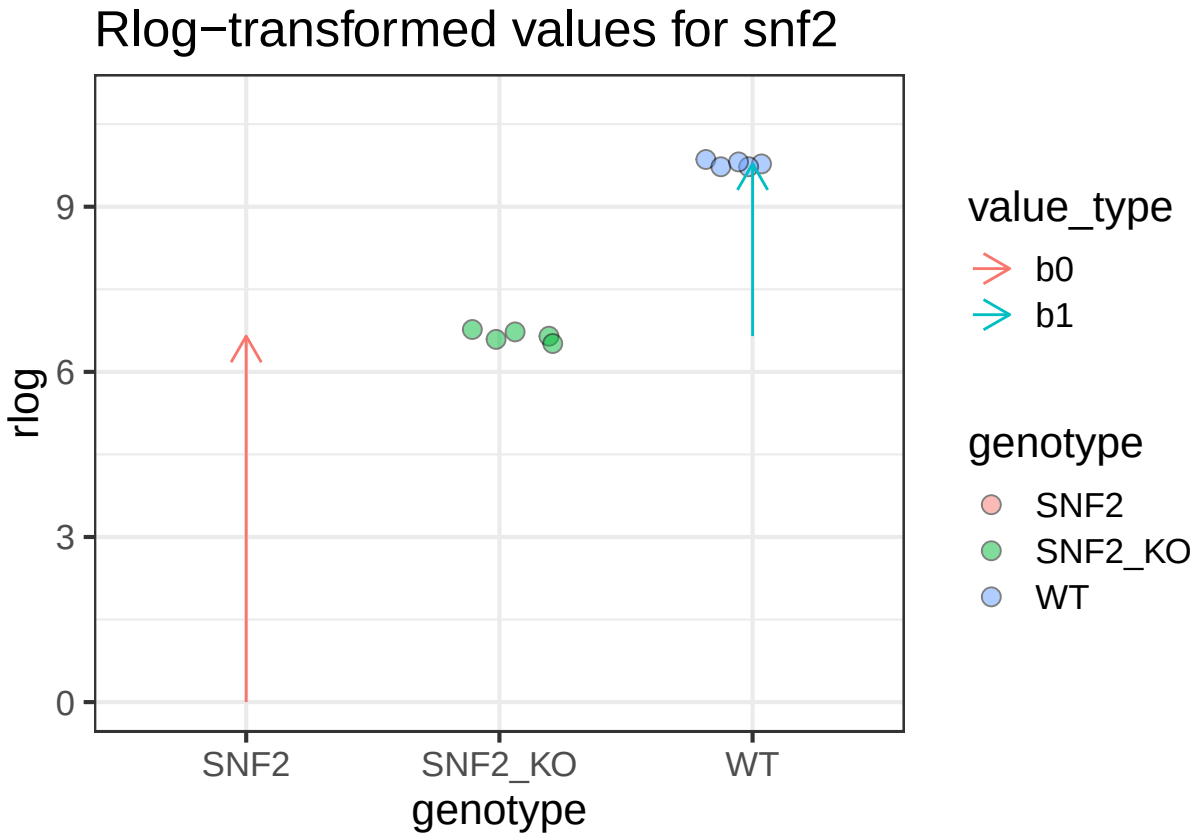
```
P <- ggplot(snf2, aes(x = genotype, y = rlog, fill = genotype)) +
  geom_jitter(width = .3, size = 3, alpha =.5, shape = 21) +
  ggtitle("Rlog-transformed values for snf2") +
  coord_cartesian(ylim = c(0, max(snf2$rlog)+1))

fitted_intercept <- coef(modelling_snf2)[1]

model_vals <- data.frame( genotype = c("SNF2", "WT"),
            value = c(fitted_intercept, fitted_intercept + coef(modelling_snf2)[2]),
            value_type = c("b0","b1"))

P + geom_segment(data = model_vals,
                 aes(x = genotype, xend = genotype,
```

```
                    y = c(0, fitted_intercept), yend = value,
                    color = value_type),
                arrow = arrow(length = unit(0.4, "cm")))
```




```
coef(modelling_snf2)
```

```
## (Intercept)  genotypeWT
##    6.649520    3.131165
```

- b0 = intercept = average of baseline group
- b1 = difference between baseline and non-reference group(s)
- b0 and b1 are actual estimates - because this is an easy example, they hit the mean of group SNF2 with b0
- both betas are estimates by minimizing the residual sum of squares
- residuals = difference between the observed values (our expression values) and the fitted ones obtained by the model when the betas take a certain value –> residuals should be as small as possible
- t-statistics: beta divided by their standard errors
- p-value = probability of finding such a larger t where H0: beta = 0
- **F-statistic** would be the one to look at because it compares the explained vs. the un-explained variance, i.e. F = between-group-variance / within-group-variance (which is the same as a t-test for a simple two group comparison)

Brief excursion: paired analysis design (also: blocked design)

For paired experimental design (e.g., different tissues from multiple patients), an additive model is recommended. Put the factor you're more interested in last. For example, if you wanted to determine genes that differ in tumor vs. healthy tissue, the following design would work with DESeq2: `design(dds) <- formula(~patient + tissue)`. The last variable (`tissue`) will be automatically used to extract the fold changes later on.

Technically, you will fit an individual base line for each patient, so that patient-to-patient differences in expression before treatment are absorbed source.

The general layout: `design <- formula(~Block+Treatment)`

Brief excursion: Accounting for possible interactions

Sometimes, you might be interested in changes where two factors are suspected to interact. Consider the example described here:

Patient.ID	Treatment	Response
1	Pre	N
1	On	N
2	Pre	N
2	On	N
3	Pre	N
3	On	N
4	Pre	N
4	On	N
5	Pre	N
5	On	N
6	Pre	Y
6	On	Y
7	Pre	Y
7	On	Y
8	Pre	Y
8	On	Y

If you wanted to see how the `Treatment` effects depend on the `Response`, you would have to add an interaction term for `Treatment * Response`.

`Treatment` effect `Treatment:Response` will then be the additional effect that comes on top of that if the patient is a responder.