Functional Genomics
Integrative analysis of next generation sequencing data

Alena van Bömmel (Alena.vanBoemmel@molgen.mpg.de R 3.3.08)
Robert Schöpflin (schoepfl@molgen.mpg.de 3.3.15)
Max Planck Institute for Molecular Genetics

Prediction model

To build the prediction model, follow these steps (recommended in R)

1. Get the promoter coordinates for all protein coding genes as Genomic Ranges, use 2000bp upstream, 500bp downstream from TSS
2. Count the reads of the histone modifications falling into these promoter regions using `bamsignals` (function `bamCount`)
3. Count the reads of the control expriments in the same way
4. Normalize the signal for the histone modifications (see slie #4)
5. Create the feature matrix X out of the log-normalized, scaled histone modification counts on promoters
6. The dependent variable Y are then the RPKM (FPKM) values for the corresponding genes (gene length [as a sum of exon lengths] needed) Alternatively, you could use the normalized read counts from the `DESeq2` object, after calling the `DESeq` function (with `counts(dds, normalized=TRUE)`)
7. Merge the replicates for the RNA-seq in a median value

Prediction model

8. Split the data into a training(50%), validation(25%) and test set(25%)
9. Build a linear regression model $Y=X*\beta + \varepsilon$ using the training data set in only one stage first (function `lm` in R)
10. Investigate the estimated β coefficients, which HM have the largest one?
11. Alternatively, use an elastic net model wit regularization
12. Calculate the R^2 statistics and the correlation between true Y and estimated Y (on training and test set)

Normalization

- estimate first the slope of the correlation between the read counts of the sample (S) versus the read counts of the input control(C) - adding a pseudo count of 1- by the median: `m=median((S+1)/(C+1))` over all promoters
- the normalized read counts are then calculated by:
- `Snorm = (S+1)/(C+1) * 1/m`
- Take the logarithm of base 2 (or 10) of the normalized counts
- Scale the log-normalized counts to be centered around 1 and scaled by the standard deviation (function `scale` in R)
- the log-normalized and scaled counts can be then used as features in the linear model