EC500: Computational Synthetic Biology For Engineers

Homework 2
Spring 2017

March 14, 2017

Genetic Design of Experiments

Due: 04/14/17, 11:59 pm

1 Overview and Background

Engineered biological systems are complex and sensitive to the tuning of many interdependent variables. By varying the genetic parts that make up these systems, we can change system behavior and screen or select for the system variants with the best performance. In practice, however, it is prohibitively expensive to build and test millions of variants to determine which ones are the best. Fortunately, there is a branch of applied statistics known as design of experiments (DOE) that addresses this challenge. DOE enables us to develop empirical models of system behavior to guide the process of design and avoid wasting resources on building and testing unnecessary system variants.

In DOE, a factorial design is a set of designs that tests how one or more "factors" affect a system of interest when taking on different combinations of quantitative "levels." In the context of synthetic biology, one could view a library of genetic constructs as a factorial design that tests how the genes of a system affect its behavior when their expression is varied. For example, one could build a library of biosynthetic pathway variants that varies the expression of genes in the pathway by placing them under the control of different combinations of genetic parts, such as promoters and terminators.

Unlike a genetic circuit in which most genes code for transcription factors, each gene in a biosynthetic pathway codes for a protein that functions as an enzyme. As shown in Figure 1, these enzymes catalyze the set of chemical reactions necessary to transform one type of organic molecule into another.

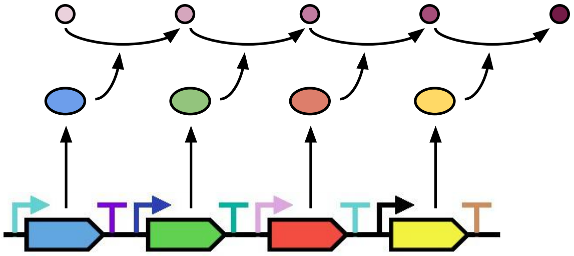


Figure 1: Schematic of a biosynthetic pathway.

For this homework, you will work in teams of two. Your goal is to optimize the design of a biosynthetic pathway so that it provides maximum yield of a desired product. In doing so, you will work through multiple iterations of a design-build-test cycle, one in which you will (1) use the Double Dutch web application to design a library of biosynthetic pathway variants based on a factorial design, (2) submit your library design and receive simulated test data based on its structure, and (3) use existing statistics software packages (such as R) or write your own scripts/software to analyze these test data and inform subsequent rounds of library design. The following sections provide an overview and additional background on each step in this design cycle.

2 Designing Libraries with Double Dutch

Designing a library of biosynthetic pathway variants based on a factorial design involves three challenges. The first challenge is categorizing and grouping DNA parts into "factor" modules for assignment to the design's factors and "level" modules for assignment to the design's levels. Large data sets can contain information on hundreds to thousands of genetic part combinations that must be categorized in this way.

The second challenge is matching level modules to the levels of the design. To facilitate their reuse across different disciplines, factorial designs are typically written in a canonical form that represents levels using small integer values that are not always representative of genetic part parameters. Hence, most factorial designs must be recalculated to make it possible to match part parameters to these designs' levels.

The third challenge is the incorporation of experimental concerns into the process of module assignment. For example, due to the risk of homologous recombination, DNA parts above a certain sequence length often cannot be reused in a pathway. This risk favors assigning level modules that contain a large variety of parts. Reagent costs and handling considerations, on the other hand, favor assigning level modules that contain the fewest unique parts possible. In the extreme case, all possible level module assignments must be evaluated to find the assignment that optimizes all of these competing concerns.

2.1 Double Dutch workflow

Figure 2 illustrates the overall workflow for designing libraries of biosynthetic pathway variants in Double Dutch. This workflow consists of three major steps: applying a formal grammar to categorize a DNA part library into factor modules and level modules, using k-means clustering to group the level modules for level matching, and performing simulated annealing to assign the level modules to the levels of a factorial design and optimize this assignment with respect to level matching, pathway homology, and DNA synthesis.

To engage with the Double Dutch workflow, simply follow these steps:

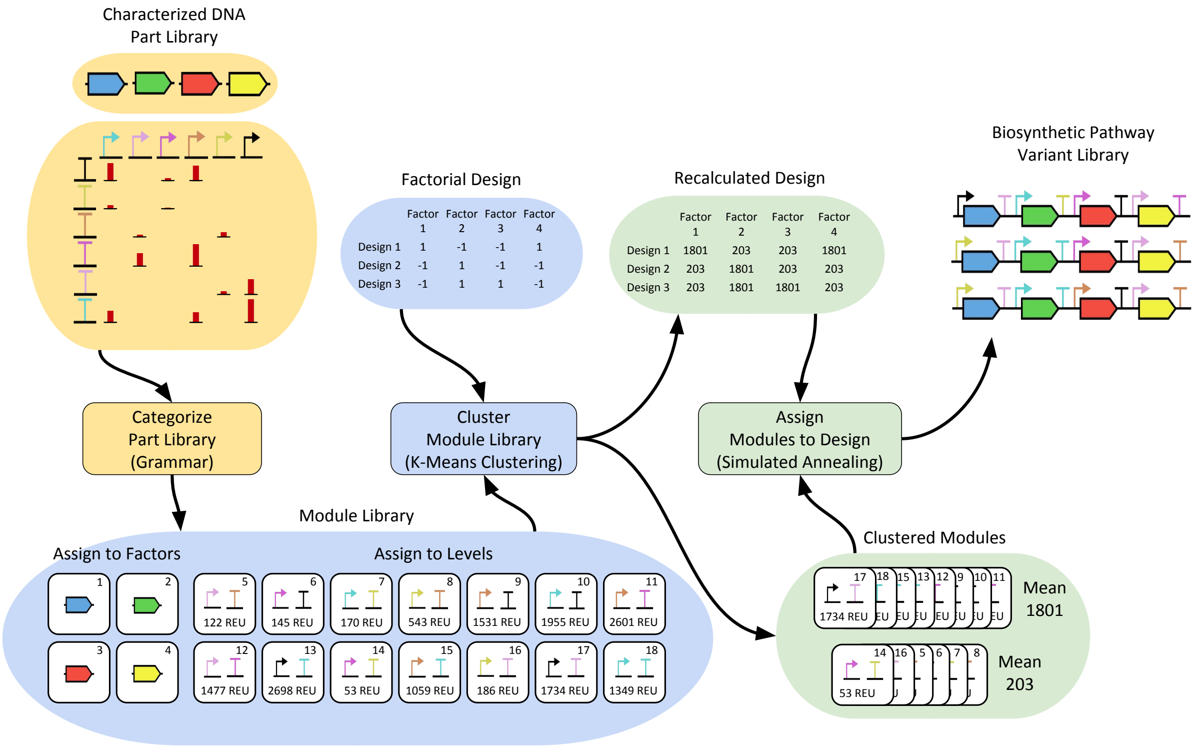
1. Upload data on your DNA part library in the form of a Comma-Separated Value (CSV) file. The CSV file for this homework is named `grid.csv` and can be found at

   `www.doubledutchcad.org`

   under the Downloads tab. Alternatively, as shown in Figure 3, you may proceed directly to the Pathway Designer application page and click the Load Example button.

2. Select the factor modules containing your pathway genes of interest and drag them from the lefthand column to the center column.

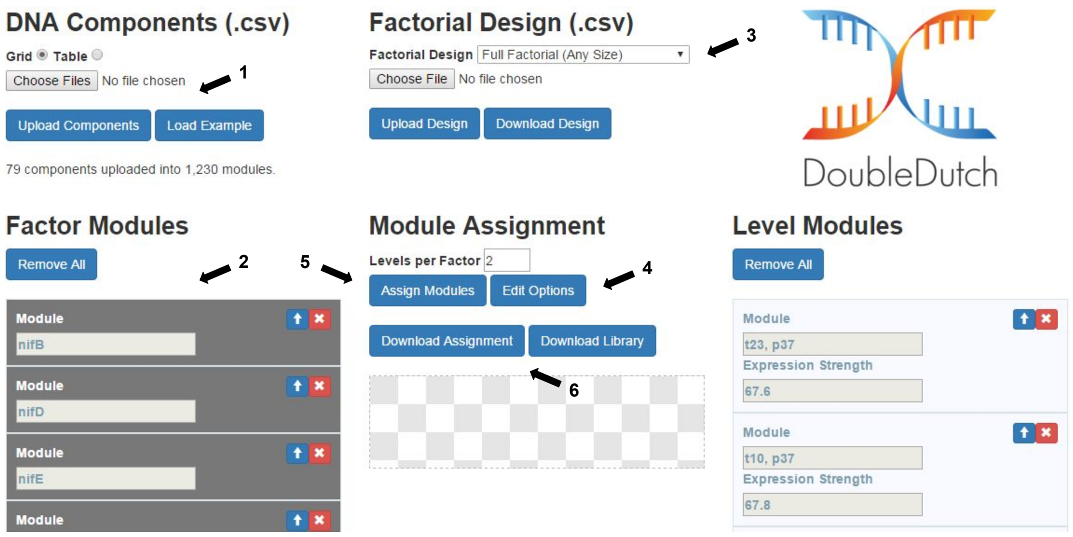
3. Select the type of factorial design on which to base your library design from the drop-down menu. Reasons to choose one type of design over another will be discussed later



	Factor 1	Factor 2	Factor 3	Factor 4
Design 1	1	-1	-1	1
Design 2	-1	1	-1	-1
Design 3	-1	1	1	-1

	Factor 1	Factor 2	Factor 3	Factor 4
Design 1	1801	203	203	1801
Design 2	203	1801	203	203
Design 3	203	1801	1801	203

Figure 2: Workflow for designing libraries of biosynthetic pathway variants with Double Dutch. Listed in parentheses under each step of the workflow is the method by which that step is carried out. The input data to each step are highlighted with the same color as that step. Each coding sequence (CDS), promoter, and terminator is represented using a symbol from the SBOL Visual standard, while the expression levels associated with pairings of specific promoters and terminators are represented using bar graphs or parameters with relative expression units (REU). In this example of the workflow, once the level modules are clustered, the mean parameter values of the resulting clusters become the new low and high levels of the factorial design. The level modules are then assigned by Double Dutch from their clusters to the corresponding levels of the design. Factor modules, on the other hand, are assigned by the user to the factors of the design before clustering (step not listed here). The final result of module assignment is a library of biosynthetic pathway variants based on a factorial design.



Grid ● Table ○

No file chosen

79 components uploaded into 1,230 modules.

Factorial Design: Full Factorial (Any Size)

No file chosen

Levels per Factor: 2

Module: nifB

Module: nifD

Module: nifE

Module

Module: t23, p37

Expression Strength: 67.6

Module: t10, p37

Expression Strength: 67.8

Figure 3: Screenshot labeled with each step in the Double Dutch workflow.

in Section 4. If you choose Full Factorial (Any Size), then you also must select the number of levels per factor.

4. Click the Edit Options button to choose your weights for the module assignment cost function and parameters for simulated annealing (these will be discussed shortly).

5. Click the Assign Modules button to run Double Dutch and obtain a level module assignment.

6. After assessing the cost of the level module assignment, you may click the Assign Modules button to run Double Dutch for another n trials, keeping only the best assignment. You may also click the Download Assignment button to obtain a CSV file that encodes the current module assignment, or click the Download Library and Library Levels buttons to obtain CSV files that contain the DNA parts and expression levels for each gene in the library of pathway variants, respectively.

7. If you wish to change the number of factor modules in the assignment or edit certain assignment options, then click Done to quit the current assignment.

2.2 Cost function

The Double Dutch cost function allows users to specify the relative importance of level matching, homology, and DNA synthesis as design concerns. Users may alter the weights of each design concern, even down to zero if they do not want Double Dutch to optimize that concern. Ultimately, changing the weights of the cost function results in different assignment costs for each design concern and different libraries of pathway variants. Figure 4 illustrates this concept for the assignment of modules from this homework's data set to a 4-factor Plackett-Burman design. Ordinarily, the assignment costs in this figure would be normalized between 0 and 1, weighted, and summed, but the raw costs are shown here to motivate a more intuitive understanding of how different weightings affect the cost distribution.

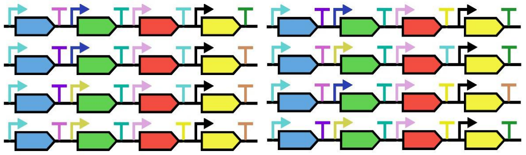
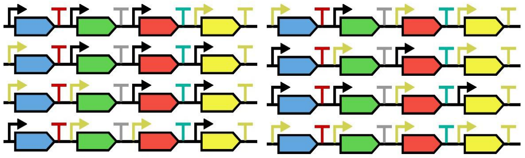
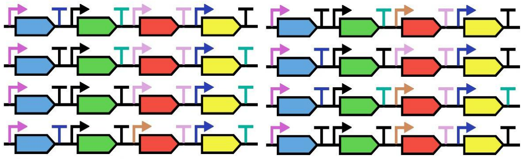
Cost Function Weights			**Raw Assignment Costs**			
Level Matching	DNA Synthesis	Pathway Homology	Total Level Difference	# of Unique, non-CDS Parts	# of Intra-Pathway Part Repetitions	**Optimized Library of Biosynthetic Pathway Variants**
1	1	1	1487.6	12	0	
1	2	1	3235.12	6	17	
4	1	1	819.94	9	7	

Figure 4: Example of how changing the cost function weights results in different assignment costs for each design concern and different pathway libraries based on a 4-factor Plackett Burman design. The first three columns list each cost function weight by the design concern that it affects, while the second three columns list the resulting raw (unweighted, non-normalized) assignment cost of each design concern. Lastly, the third column displays the libraries of biosynthetic pathway variants resulting from each module assignment.

In this example, doubling the DNA synthesis weight halves the number of unique, non-CDS DNA parts in the final library, but results in 17 part repetitions in the pathway variants (only repetitions within individual pathways count) and doubles the total difference between the parameters of the assigned modules and the levels of the factorial design. Note in this case Double Dutch assigns only the black promoter or olive promoter to every gene in every pathway variant. Quadrupling the level matching weight, on the other hand, roughly halves the total difference between parameters and levels, but results in 7 part repetitions in the pathway variants (the black promoter and black terminator). The number of unique, non-CDS parts in the library is also decreased by a quarter, but this is expected given that DNA synthesis and pathway homology are partly in opposition as design concerns. In both cases, optimization comes at the expense of increasing the assignment cost of at least one other design concern.

2.3 Simulated annealing

The parameters of the Double Dutch simulated annealing heuristic include a number of trials n, a number of iterations per trial k, an initial temperature t', and a constant b. During each trial of simulated annealing, a total of k "mutations" or module swaps are made to an initially random module assignment. If a given mutation lowers the assignment cost, then it is accepted. Otherwise, the mutation is accepted with the following probability:

$$\mathbb{P}(accept) = e^{b(t'/t)(s-s')} \tag{1}$$

In Equation 1, s is the pre-mutation assignment cost and s' is the post-mutation assignment cost. Accepting worse assignments in this manner can help prevent the search for the assignment with the smallest cost from stopping at a local minimum. After each mutation, the current temperature t is multiplied by a factor of $(1/t')^{1/k}$, thereby decreasing t and the probability of accepting worse module assignments over time until none are accepted. The parameters t' and b influence how this probability changes with decreasing t, with b also determining the range of initial probabilities.

Finally, the worst-case runtime for simulated annealing in Double Dutch scales as the product of n, k, and the size of the design (the sum of the number of levels that each of its factors may take on). The latter term exists because each mutation in a given trial results in the recalculation of the assignment cost, which has a runtime that depends on the size of the factorial design.

3 Testing Library Designs

In the real world, constructing and testing a library of biosynthetic pathway variants can take weeks. Fortunately, we will be simulating this process to obtain artificial test data on your library designs. In order to test a library design, you will e-mail the CSV files obtained by clicking the Download Assignment, Library, and Library Levels buttons in Double Dutch to `nicholasroehner@gmail.com`. You are guaranteed to receive test data on a library by 9 am at the latest if you e-mail the appropriate CSV files before 11 pm on the previous day. If you wish to generate the CSV files using a custom software tool, make sure that they match the format of those generated by Double Dutch.

The format of the test data you receive will match that of the library level CSV file, with the exception that there will be two additional columns of real numbers. The first column will have the header "yield" and will document the average concentration of product in mg/L that is obtained when testing each pathway variant in triplicate (three different samples). The second column will have header "yieldSD" and will document the standard deviation of each such test. Note that the second additional column is not required for linear regression and is included just in case you wanted to know this information.

There are two points to consider before submitting a given library design for testing. The first point is, for the purpose of this homework, testing each library costs a number of "credits" equal to the total number of unique DNA components (promoters, CDSs, and terminators) among its pathway variants, plus one credit per pathway variant. For example, the first library of pathway variants in Figure 4 would cost 24 credits to test (4 CDSs + 5 promoters + 7 terminators + 8 pathway variants). Each team starts the homework with 500 credits. Once you have used up your credits, you will be unable to test any more library designs, so think carefully about your designs and make them count!

The second point deals with pathway homology. You are welcome to submit library designs in which individual pathway variants contain repeated DNA components, but there is a chance that that these variants will fail and not provide any data during testing. In addition, your final optimized pathway design must not contain any repeated components (see Section 5).

4 Analyzing Test Data

Once you have collected test data, you can perform a regression analysis to relate a response variable y to a set of control variables $x_1, x_2, \ldots, x_k$. For this homework, we are interested in modeling the relationship between the yield of a biosynthetic pathway and the expression levels of genes in this pathway. More specifically, we will assume that this relationship can be modeled by the following equation:

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \sum_{j=1}^{i} \beta_{ij} x_i x_j \tag{2}$$

In this model, the intercept β_0 captures pathway yield in the absence of gene expression. Furthermore, each coefficient β_i captures the main effect of an individual gene's expression on pathway yield, while each coefficient β_{ij} captures any interaction effect between a pair of genes' expression levels. For this homework, we are assuming that interaction effects between a gene's expression and itself do not occur.

Once you have collected test data on your library design, it is common in classical DOE to fit the data to this type of model using multiple linear regression analysis. Before you perform either of these steps, however, you should consider the factorial design that the library was based on . This is because the type of factorial design used to collect data has implications for which terms in the model may be biased due to aliasing. Aliasing can make it impossible to distinguish between the main and/or interaction effects of different factors by inflating their coefficients.

4.1 Factorial designs

The four types of factorial designs that are available by default in Double Dutch are full factorial designs, Box-Behnken designs, 2^{k-p} fractional factorial designs, and Plackett-Burman designs. Table 1 lists these designs and their relevant properties, which are briefly described here and in suggested reading #1.

Table 1: Types of Factorial Designs Available in Double Dutch

Name	**# of Factors**	**# of Levels**	**# of Designs**	**Resolution**
Plackett-Burman	$N < 24$	2	Multiple of $4 \geq N + 1$	III
2^{k-p} Fractional Factorial	N	2	Power of $2 \geq N + 1$	III, IV, V
Box-Behnken	$N < 13$	3	Fraction of 3^N	N/A
Full Factorial	N	M	M^N	N/A

A full factorial design tests all possible combinations of levels that its factors can take on. While full factorial designs allow us to estimate the coefficients of Equation 2 without aliasing bias, they can be inefficient and costly to implement owing to their large size. A Box-Behnken design also lacks aliasing bias with respect to Equation 2, but is much more economical than a full factorial design by virtue of testing a fraction of all possible level combinations that its factors can take on.

Lastly, the Plackett-Burman and 2^{k-p} fractional factorial designs are the among the most economical factorial designs, but this economy comes at the expense of aliasing the main effects and/or interaction effects of these designs' factors. The degree of aliasing is commonly stated as the resolution of a design and is inversely proportional to the size of the design. Double Dutch is capable of generating 2^{k-p} fractional designs of resolution III, IV, and V, which have the general aliasing patterns described below (Plackett-Burman designs are resolution III).

1. **Resolution III:** Main effects may be aliased with two-factor interaction effects. We will discuss in class how to tell which effects are aliased. For Plackett-Burman designs, the main effect of each factor is partly aliased with the effects of all two-factor interactions that do not involve that factor.

2. **Resolution IV:** Main effects are not aliased with two-factor interaction effects, but two-factor interaction effects may be aliased with other two-factor interaction effects.

3. **Resolution V:** Main effects are not aliased with two-factor or three-factor interaction effects, and two-factor interaction effects are not aliased with other two-factor interaction effects. Two-factor interaction effects may be aliased with three-factor interaction effects, but the latter are often assumed to be negligible.

Figure 5 illustrates one possible strategy to take when choosing a series of factorial designs for designing libraries of pathway variants and optimizing pathway yield. Since it is entirely possible that one or more in genes in the pathway have little to no effect on its yield, this strategy starts with a less costly Plackett-Burman or a low resolution (III or IV) 2^{k-p} fractional factorial design to identify the genes that do have significant main effects, ignoring for the moment any interaction effects that may occur between genes. Once the statistically insignificant terms corresponding to ineffective genes have been dropped from the regression model, we can follow up with a more costly Box-Behnken or resolution V fractional factorial design to identify if there are any significant interaction effects. Finally, we can use this high-resolution regression model to determine the expression levels for each gene that maximize pathway yield (see Subsection 4.3), then run additional high-resolution designs centered on these points and perform additional regressions to determine the selection of DNA parts that optimize pathway yield.

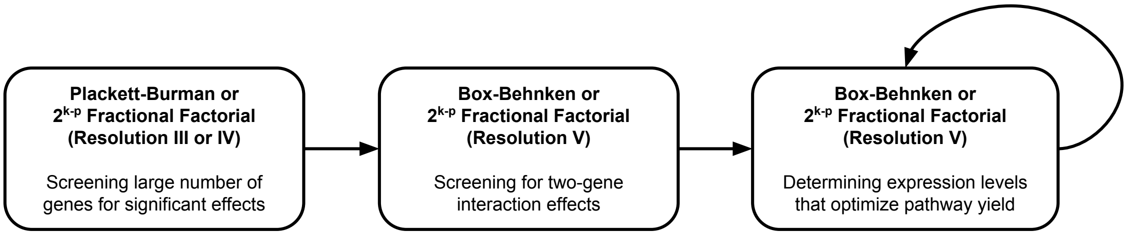

Plackett-Burman or
2k-p Fractional Factorial
(Resolution III or IV)
Screening large number of genes for significant effects
Box-Behnken or
2k-p Fractional Factorial
(Resolution V)
Screening for two-gene interaction effects
Box-Behnken or
2k-p Fractional Factorial
(Resolution V)
Determining expression levels that optimize pathway yield

Figure 5: Example of strategy for choosing factorial designs to optimize pathway yield.

4.2 Multiple linear regression

There are many different software packages that can be used to perform a multiple linear regression, but for this handout we will focus on R. The R script that follows performs a regression on the test data file "mydata.csv," modeling the relationship between pathway yield and the expression levels for the genes nifB, nifD, nifE, nifH, nifK, and nifM. Note that interactions can specified with the syntax `nifB:nifD`. We will cover this and other useful R commands in class—see also suggested readings #2 - 3.

```
mydata <- read.csv("yourpathhere/mydata.csv")
fit <- lm(yield ~ nifB + nifD + nifE + nifH + nifK + nifM)
summary(fit)
```

Table 2 is an example of R's output when the `summary` function is run. There at least three questions that we should ask ourselves about this output.

1. **Are the coefficients of the regression model statistically significant?** This can be determined by looking at the t value of each coefficient and its accompanying p-value

Table 2: Example of Multiple Regression Results

```
Coefficients:
                 Estimate  Std.  Error  t value   Pr(<|t|)
(Intercept)    -91.928601   20.531195   -4.478  4.80e-05    ***
nifB             0.008922    0.002040    4.374  6.74e-05    ***
nifD             0.002393    0.002130    1.123   0.26702
nifE             0.005158    0.001715    3.008   0.00421     **
nifH             0.006453    0.002270    2.842   0.00660     **
nifK             0.004787    0.001883    2.543   0.01436      *
nifM             0.006132    0.002386    2.571   0.01338      *
---
Signif.  codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'  0.1 ' ' 1
Residual standard error:  51.29 on 47 degrees of freedom
Multiple R-squared:  0.5063, Adjusted R-squared:  0.4433
F-statistic:  8.033 on 6 and 47 DF, p-value:  5.426e-06
```

(right), which is the probability of observing the test data given the null hypothesis that the coefficient is equal to zero. If the p value is less than 0.05, then we can be fairly confident in rejecting the null hypothesis and asserting that the coefficient is not equal to zero. Coefficients that are not significant should not be included in your regression model.

2. **Does the regression model predict the test data well?** This question is partly answered by looking at the R^2 value for the regression. A large R^2 indicates that the test data for the dependent variable (in our case, pathway yield) are close to the predictions of the regression line, with $R^2 = 0$ indicating no fit and $R^2 = 1$ indicating a perfect fit. R^2, however, is not the only metric that you should use in assessing whether the regression model is a good fit, nor is it the best one. Adding terms to the model will always cause R^2 to increase regardless of whether these terms are good predictors of the test data, a problem known as overfitting. When adding terms, you should look to see whether the adjusted R^2 value increases significantly, as this metric takes into account the number of terms in the model and will not always increase with each additional term.

3. **If so, do the residuals of the regression model support your conclusion?** Residuals are the differences between the test data and the predictions of the model, with a positive residual indicating that the model has underestimated the test data and a negative residual indicating that the model has overestimated the test data. Residual plots are another important tool for assessing the fit of the regression model and tell us if our test data is nonlinear, heteroscedastic, or contains outliers. In the first two cases, it may be sufficient to transform the test data for the dependent and/or

independent variables using log base 10, square root, or another nonlinear function to approximate a linear, homoscedastic system as assumed by linear regression. You will learn more about residual plots and other considerations for regression analysis in class and suggested readings #4 - 5.

4.3 Model optimization

Once you have obtained a regression model, the next step is to optimize the model by determining the values of the independent variables (gene expression levels) that the model predicts will produce the maximum value of the dependent variable (pathway yield). As previously discussed in Subsection 4.1, these values can then inform the selection of a new factorial design and its implementation as a new library of pathway variants.

Regression models can be optimized using a variety of techniques, such as the method of ridge analysis discussed in suggested reading #1 or a simulated annealing heuristic similar to that used in Double Dutch. It is also possible that a much simpler strategy (such as using the strongest parts) that does not require automation via a script or program would suffice. Ultimately, the choice of how to optimize your regression models is left to you. What matters most is that you report which methods you use and explain your reasoning in choosing them.

5 Scoring

120 points: Total points for Homework 2

- **120 points:** Write a report (4 to 6 pages, including references) that includes the following sections:
  - **20 points:** Methods section that describes your strategy for choosing factorial designs (what type, what order, and why) and your strategy for optimizing regression models (ridge analysis, simulated annealing, or other and why).
  - **60 points:** Results section that discusses at least three of your library designs. Each discussion should focus on the fit of your chosen regression model to your library data (adjusted R^2, significance of terms, residual plots, and why that model versus others) and how your chosen model influenced your subsequent round of library design (which genes have an effect, which gene expression levels are predicted to optimize pathway yield, and how these affect your next library design).
  - **30 points:** Conclusion section that describes your final pathway design (genes and the promoter-terminator combinations controlling them, no part repetition) and the yield of this pathway when tested. Explain the reasoning behind your design.
  - **10 points:** References for methods used (can include suggested reading)

6 Submission

E-mail a PDF copy of your report to `nicholasroehner@gmail.com`. Questions about the assignment can also sent to this address.

7 Suggested Reading

1. Khuri, A. I. and Mukhopadhyay, S (2010). Response surface methodology. WIREs Computational Statistics, 2, 128-149.
   Available at `www.stat.ufl.edu/personnel/usrpages/RSM-Wiley.pdf`
   Part I of this paper covers the factorial designs available in Double Dutch (among others) and how they can be used in DOE.

2. Kabacoff, R. I. Quick-R. Available at `www.statmethods.net`.
   This is a helpful guide to R with examples of using its many functions.

3. Model selection in R.
   Available at `http://www2.hawaii.edu/∼taylor/z632/Rbestsubsets.pdf`.
   This is documentation for the R package `leaps` and its principal function `regsubsets`. We will discuss in class how this function can be used to perform many simultaneous multiple regressions and compare them.

4. Frost, J. Regression analysis tutorial and examples.
   Available at `blog.minitab.com/blog/adventures-in-statistics/regression-analysis-tutorial-and-examples`
   This is collection of blogs that discusses regression analysis for users of the Minitab software, but many of its points are relevant regardless of the software package that you are using.

5. Holland, S. Regression diagnostic plots.
   Available at `strata.uga.edu/6370/rtips/regressionPlots.html`
   This class material provides great examples of residual plots and their interpretation (and code to generate these examples).