10.007 Systems World 2D Project

Due 11pm, Sunday 21 April 2019

Part A – curve fitting on PCR data

Polymerase chain reaction (PCR) is a technique used in biology to amplify segments of DNA. The technique exposes the reactants to cycles of repeated heating and cooling. Through the use of fluorescent dyes, the amount of amplified DNA product can be measured after each cycle by its fluorescence. However, due to factors such as background noise, the measurements are imperfect. Your goal here is to enable accurate determination of the amount of product, by fitting a smooth curve to some fluorescence vs cycle data (see *Excel* file).

Question A1 (warm up): for cycles (C) 8 to 18, the fluorescence (F) can be modeled by an exponential curve,

$$F = \alpha\, e^{\beta C}.$$

Linearize this model, then use *regression* to estimate α and β. You need to show working and you may not use *Excel*'s Trendline function.

To model the fluorescence for all the given cycles (8 to 36), we use a more complicated curve,[1]

$$F = \frac{M}{1 + e^{-k(C-d)}}. \tag{1}$$

It is harder to linearize this model and to estimate the parameters M, d and k. We proceed as follows.

Let C_i be the ith cycle and F_i the ith fluorescence in the given data (so $C_1 = 8$, $F_1 = 8.2$, etc). For equation (1) to be a good approximation to the data, we must have

$$F_i \approx \frac{M}{1 + e^{-k(C_i-d)}}, \quad \text{for} \quad i = 1, 2, \ldots, 15. \tag{2}$$

Question A2: estimate M and d from the graph. (Hints: M is the 'plateau position'; consider what happens if $C_i = d$.)

With M and d estimated, the only unknown parameter is k. To estimate k, we can *linearize* equation (2) and define an *error* term ϵ_i, which has the form

$$\epsilon_i(k) = a_i\, k + b_i, \tag{3}$$

where a_i and b_i only depend on the data points (C_i, F_i), M, and d.

Question A3: determine a_i and b_i in equation (3); show working.

We would like to pick k such that the *total absolute error*, given by

$$\sum_{i=1}^{15} \left|\epsilon_i(k)\right|,$$

is *minimized*. That is, we wish to solve the optimization problem $\mathcal{P}$:

$$\left.\begin{array}{rrl} \min & \displaystyle\sum_{i=1}^{15} \left|\epsilon_i(k)\right| & \\ \text{subject to:} & k & \geq\ 0 \end{array}\right\}$$

[1] It takes into account the leveling off of the reaction rate, due to reasons such as the reactants being consumed.

However, the objective function in $\mathcal{P}$ is *not* a linear function of k, due to the absolute value sign. We must find a way to linearize the problem.

Question A4: let f be a linear function of k. Then the optimization problem

$$\min\ |f(k)|$$

can be written as a *linear program*,

$$\left.\begin{array}{rcll} \text{min} & z & & \\ \text{subject to:} & & & \\ \text{(constraint 1)} & z & \geq & \text{expression 1} \\ \text{(constraint 2)} & z & \geq & \text{expression 2} \end{array}\right\}$$

Find the expressions in the two constraints.

Question A5: generalize your answer to Question **A4**, and hence rewrite the problem $\mathcal{P}$ as a linear program. Then, implement and solve it in *Excel*, using the 'Simplex LP' method, to find a good estimate for k.

You may use values of M and d that are different from what you obtained in Question **A2**, if they give a smaller total absolute error. In your answer, you need to include your LP in *Excel*, and report on your best values of M, d, k, and the total absolute error (you will be assessed on how small you can make it).

Part B – primer design using ILP

Specifically, PCR amplifies a segment of DNA between a forward primer and a reverse primer. Each primer is a short strand of DNA, and needs to satisfy some constraints. It turns out that these constraints can be modeled by an *integer linear program*, and so any feasible solution to such an ILP provides a viable primer.

Question B1 (warm up): write down an integer linear program whose solution corresponds to a 'toy' primer, where:

- The primer has 10 bases.
- Each base is either an `A` or a `C`.
- The number of `C`'s is between 4 and 6 (inclusive).
- Among the last 5 bases, there are 2 `C`'s.

Use *Excel* `Solver` to find a primer that satisfies the above constraints.

A *reverse* primer is given in the *Excel* file. Your next task is to find a *forward* primer that is compatible with it. Such a primer needs to meet the constraints given below.

Question B2: write an integer linear program in *Excel* which designs a *forward* primer, satisfying these constraints:

- The *forward* primer has 18 bases,
- Each base is either an `A`, a `C`, a `G` or a `T`.
- The total number of `C`'s and `G`'s in the *forward* primer is between 7 and 11 (inclusive).
- The total number of `C`'s and `G`'s in the last 5 bases is at most 3.
- Consider the quantity (4 times the number of `C`'s and `G`'s) + (2 times the number of `A`'s and `T`'s) for each of the *reverse* and the *forward* primer. The two numbers obtained should equal.[2]

[2] This is to ensure that both primers have similar annealing temperatures.

In your answer, you need to include your ILP in *Excel*, and identify a primer, found using `Solver` under the 'Simplex LP' method, which satisfies these constraints.

The primer found in Question **B2** may still not be ideal, because there are further criteria that a good primer should satisfy.

Question B3: additionally, model as many of the following constraints as possible:

- The *forward* primer has no 'runs' of length more than 4. A run is a consecutive sequence of the same base, for example `AGGGCG` contains a run of 3 `G`'s.
- The *forward* primer has no alternating patterns of the form XYXY. For example, `AGCGCT` contains such an alternating pattern in the middle.[3]
- When the *forward* and *reverse* primers are aligned side by side, the number of alignments between `A`'s from one primer and `T`'s from the other is less than or equal to 4; the same goes for `C`'s and `G`'s.[4] For example, there are 3 `A-T` alignments between `ACTTT` and `TGCAA`.

Excel `Solver` has a limit of 200 variables and 100 constraints, so not all of the above constraints can be implemented at the same time in your ILP. You should implement as many as you can in your final program, and clearly explain how to model the ones that you know how to do but did not implement; alternatively, you can use OpenSolver to implement all the constraints.

Bonus question: instead of treating the reverse primer as a constant, now treat it as a *variable*, so your *Excel* output should be a pair of primers (forward and reverse) satisfying all the constraints in Questions **B2** and **B3**. Since this increases the number of decision variables, you may wish to use OpenSolver.

Submission instructions

- Work in your assigned 2D groups.
- Each group is to submit a single file (e. g. a zip) by emailing it to both of their cohort instructors before the deadline. Late submissions will incur a penalty. The file should be named F0X_GroupY_2D (with the appropriate X and Y). The file should clearly show the name of each group member who contributed to the project.
- The file should include: your written/typed up answers to Part **A**; an *Excel* linear program for **A5**; your written/typed up answers to Part **B**; all the *Excel* integer linear programs for **B1**, **2**, **3**.
- The non-*Excel* part of your submission has a page limit of four A4 sized pages. All *Excel* files should be *annotated* (e. g. contain explanations on how you modelled each constraint) so that a reader can easily reconstruct your reasoning.
- If you receive help from anyone outside your group, or consult online material or books, you **must** give them appropriate attribution. You **must** write your own solutions. Groups found plagiarizing other people's work will be given a score of 0.
- All students will be required to complete a peer evaluation survey; this is to ensure that all group members are accountable for their contributions.

[3] This and the previous constraint are required to avoid mis-priming.

[4] This is to prevent primer dimers.