

MOL518 Spring 2026

Homework 1

Due: Friday February 6, 2026

For this assignment, you will generate and then submit a jupyter notebook on canvas. You may use generative AI tools as per the syllabus unless otherwise stated in the problem.

First, make a new Jupyter notebook named `MOL518_HW1_<YourLastName>.ipynb`. All problems will be answered in this notebook. For each problem, start with a Markdown cell using Problem XX as the title, i.e. “# Problem 1”.

Problem 1: Practice with Markdown

Figure 1 is the preface page for Lubert Stryer's classic textbook *Biochemistry*. Recreate this page in a Markdown cell in your notebook. The image at the bottom of the page is provided as `StryerPreface_liverketone.png`. *Do not use AI for this problem.*

PREFACE

The more we learn, the more we discover connections threading through our biochemical world. In writing the sixth edition, we have made every effort to present these connections in a way that will help first-time students of biochemistry understand the subject and how very relevant it is to their lives.

Emphasis on Physiological Relevance

Biochemistry is returning to its roots to renew the study of its role in physiology, with the tools of molecular biology and the information gained from gene sequencing in hand. In the sixth edition, we emphasize that an understanding of biochemical pathways is the underpinning for an understanding of physiological systems. Biochemical pathways make more sense to students when they understand how these pathways relate to the physiology of familiar activities such as digestion, respiration, and exercise. In this edition, particularly in the chapters on metabolism, we have taken several steps to ensure that students have a view of the bigger picture:

- Discussions of metabolic regulation emphasize the **everyday conditions** that determine regulation: exercise versus rest; fed versus fasting.
- New **pathway-integration figures** show how multiple pathways work together under a specific condition, such as during a fast.
- More **physiologically relevant examples** have been added throughout the book.

This physiological perspective is also evident in the new chapter on drug development. The use of a foreign compound to inhibit a specific enzyme sometimes has surprising physiological consequences that reveal new physiological principles.

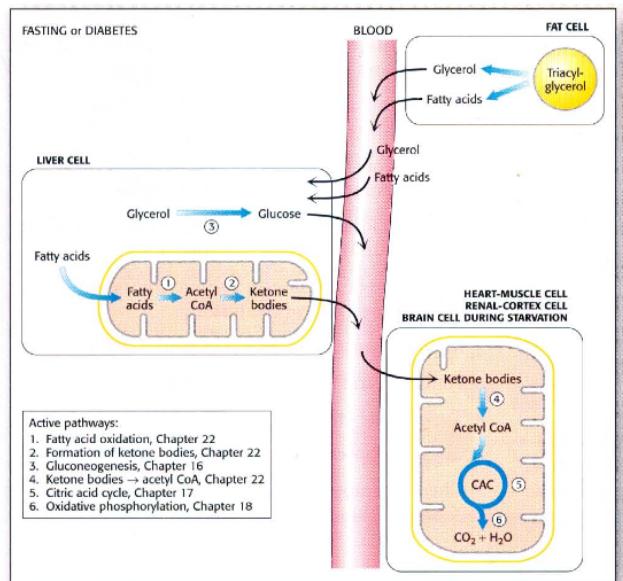


Figure 1: Stryer preface page.

Problem 2: Wet-Lab Arithmetic — MOI Setup (No Loops)

(15 points)

You are infecting cells in a single well. **Do not use for loops or if statements in this problem.**

- a) Define the following variables (with these exact names) in Python:

- `cells` = 250000 (cells)
- `moi` = 0.2 (dimensionless)
- `titer` = 2.0e8 (infectious units per mL)
- `inoc_vol_uL` = 500 (μL total inoculation volume)

- b) Compute the number of infectious units needed for this MOI:

```
iu_needed = cells * moi.
```

- c) Convert the titer to infectious units per μL , and compute the volume of virus stock to add (in μL).
d) Compute how many μL of media to add so the final inoculation volume is `inoc_vol_uL`.
e) Print a short, readable summary with your final values (infectious units, virus volume, media volume).

Problem 3: CSV “Sanity Check” Report (No Loops)

(15 points)

Load growth curve data and compute a few quick summary quantities. **Do not use for loops or if statements in this problem.**

- a) Load `Lecture_2/data/growth_curve1.csv` as a 2D NumPy array called `data`.
b) Extract `time` (seconds) and `od` (unitless) as 1D arrays.
c) Print the shape of `data`, plus the first row and last row.
d) Compute and print the sampling interval in seconds: `dt_sec = time[1] - time[0]`.
e) Compute and print the total duration in hours.
f) Compute and print:
 - absolute OD change: `od[-1] - od[0]`
 - fold change in OD: `od[-1] / od[0]`

Problem 4: Create, Transform, and Save a Synthetic Dataset (No Loops)

(15 points)

You will generate a tiny synthetic “growth” dataset (time vs. OD), then save and reload it. **Do not use for loops or if statements in this problem.**

- a) Create a NumPy array called `time_hr` with these time points (hours): [0, 1, 2, 4, 8, 16].
- b) Define `od0 = 0.08` and `doubling_time_hr = 1.5`.
- c) Compute `od` using: `od = od0 * 2**((time_hr/doubling_time_hr))`.
- d) Create `od_norm` by normalizing `od` to its first value.
- e) Use `np.column_stack` to make a 2D array with columns `time_hr`, `od`, `od_norm`.
- f) Save the result to `Homework_1/simulated_growth.csv` using `np.savetxt(..., delimiter=",")`.
- g) Reload the file you saved into `data2` and print `data2.shape` and the first 3 rows.

Problem 5: Lists — Sample Tracking and Lane Batching (No Loops)

(15 points)

You will practice list creation, indexing, slicing, appending, and sorting. **Do not use for loops or if statements in this problem.**

- a) Create a Python list called `samples` with the following sample IDs (strings), in this exact order:

```
samples = [  
    "WT_A", "WT_B", "WT_C", "K01_A", "K01_B", "K01_C",  
    "K02_A", "K02_B", "K02_C", "Rescue_A", "Rescue_B", "Rescue_C"  
]
```

- b) Print the number of samples using `len(samples)`.
- c) Append two new samples to the end of the list: "Blank" and "PositiveCtrl".
- d) Create a new list called `samples_sorted` that is an alphabetically sorted version of `samples` (do not change the original list).
- e) Create three new lists by slicing `samples_sorted`:
 - `lane1` = first 5 samples
 - `lane2` = next 5 samples
 - `lane3` = remaining samples
- f) Print `lane1`, `lane2`, and `lane3`.