CHAPTER 1

The Digital Cell Philosophy

The philosophy behind *The Digital Cell* is that *quantification* is the key to understanding cell biology. Results should be measurable, and any qualitative results should be quantified. Next, all analysis should be *automated* as far as possible. This is to remove bias with the aim of making analyses reproducible. To make all of this happen, *organization* is crucial so that links can be made between experimental design, execution, data, quantification, and results.

Experimental data are captured and stored in an organized way that allows easy access and reuse. Computer programs perform analysis of the raw data via a workflow or pipeline. The output is a readout of the analysis. This is the result and it is disposable. I will explain why, but first we need to clarify what we mean by a workflow or pipeline.

WORKFLOWS AND PIPELINES

Setting up an automated workflow or pipeline means that your analysis will be reproducible. Ideally, the number of steps involving a human is minimized because, sadly, it is we humans who introduce errors into analyses. Workflows and pipelines are also a huge time-saver. A manual workflow will take time. Imagine you get to the end and then decide a parameter needs to change; you want to measure something slightly differently or maybe you collect a new data set. This means you have to do it all again. If you have programmed a computer to do all of the intervening steps, all you have to do is click "go" once more.

Computational methods for processing data are referred to as workflows or pipelines. The two are similar in that you put raw data in at one end and get a final analysis out at the other. A pipeline is where this processing is seamless. The data are fed in and then automatically passed to other modules to achieve the final analysis, requiring no human intervention. A workflow, on the other hand, is where there are intermediate products that must be manually fed into another program or module before the final result is achieved. For example, a pipeline might take a directory of images, extract information, do some calculations, and make a plot of the data all within one software package. A workflow, on the other hand, might involve

analyzing a directory of images using a script and then feeding the output of that analysis into a separate package to produce the final plot.

Two things are implicit in the design of a workflow or a pipeline. First, raw data are treated as read-only (unchangeable). Second, outputs (graphs, figures, and so on) are disposable. The data can be fed into the workflow again and again and again. And then fed into some other workflow in the future. This means that the data stay untouched, in the raw form in which they were captured. The raw data are a separate entity from the workflow, and any other data set should be able to be fed in. This also means that what comes out is not precious. The resulting plots and files should be designed to be updated, overwritten, or otherwise just deleted. Can't remember if the outputs were done with the latest version of the workflow? No problem! Just delete them and run it again.

This means that all anyone would ever need to reproduce your analysis is a data set and the workflow/pipeline. Do not forget that "anyone" includes you! Your future self is the immediate beneficiary of organization, documentation, automation, and reproducibility.

USING SPREADSHEETS FOR EXPERIMENTAL DATA

Developing an analysis workflow or pipeline means passing data from one program to another. These files are normally long lists of numbers or words—for example, the results of analyzing the intensity of fluorescence in several cells over many time points. Your first instinct might be to put these results into a spreadsheet program so that you can look at them. Passing data from one program to another is best achieved using simple files organized for computers (and not humans) to read them. Ideally, these files are text files of comma-separated values (csv) or some other simple format. Spreadsheet programs can create these simple format files, but it is best to avoid using such programs if possible.

Spreadsheet programs such as Microsoft Excel are incredibly easy-to-use and are ubiquitous, but they are problematic for use in science. Excel is designed for compiling sales figures in a business environment. It performs poorly as a scientific application for several reasons:

- It is not auditable. Errors can be easily introduced by accident, and the user would never know. It is very difficult to find mistakes because there is no history window allowing the user to see what has happened.
- It is not good for biological data. For example, protein names such as OCT4 or SEPT9 get automatically converted to dates.
- Worksheets are limited to 1,048,576 rows and 16,384 columns.

- Excel cannot make high-quality graphics for publication.
- Users tend to compile summary statistics in the same sheet as the primary data, which prevents simple output of raw data for analysis.
- Excel lends itself to presentation of data for humans to look at but not for the organization of data frames that a computer can readily understand.

Having said all this, avoiding Excel entirely is not practical and there are areas in which Excel does indeed excel:

- It is very useful for organizing data quickly.
- It is great for a quick visual scan of the entire data set and for spotting obvious errors.
- Excel is useful for quick calculations and for preparing simple charts to show informally.
- The charts are dynamic and update on-the-fly.
- Powerful functions like `VLOOKUP()` and Pivot Tables are easy to perform, whereas they can be cumbersome to perform in other packages.
- Exporting as csv or other formats for use in another program is simple.

You need to be aware of the limitations of Excel and always organize your data with future reuse in mind:[1]

- Turn off the automatic functions that you do not need, because they might interfere with your data.
- Enter data consistently. For example, stick to YYYY-MM-DD for dates.
- Do not leave empty cells. They could mean data not collected, a mistake in entry, or 0.
- Do not do calculations among the raw data; if you want to do this, use a separate worksheet.
- Do not use coloring of cells or other formatting tricks to encode information; the information cannot be exported easily.
- Organize the data in a rectangle starting at A1.
- Save out a copy as a csv file or other delimited text file.

Outputs from image analysis software are typically in csv format. So long as the data are well organized and you use a sensible filenaming system (see Chapter 2),

these files will be all you need to analyze and present your data. With some practice you can avoid using spreadsheet programs entirely.

SOFTWARE FOR DIGITAL CELL BIOLOGY

In the book we will use two software types: ImageJ for image processing and R for number crunching. Specifically we will use Fiji and RStudio as environments for ImageJ and R, respectively. These environments are explained in detail in Chapter 3. They were selected because they are freely available, open-source, widely adopted, and likely to be around for a long time. We will also use the command line of your computer to perform powerful operations and unlock a new world of possibilities for you and your research. There are a number of scripts, macros, and code examples in the book.[2] These have been kept deliberately simple so that they can be understood. The aim is for you to build on these examples for your own work. To save rekeying the examples, they are available at https://doi.org/10.5281/zenodo.2643410.

FOCUSING ON IMAGING

Cell biology is a broad subject and encompasses many techniques—from structural biology and biochemistry to immunology and genetic analyses. It is not possible to cover in one short book all the data types and methods that you might use as a cell biologist. Instead I have concentrated mainly on imaging data from microscopy experiments. Microscopy is at the core of most cell biological studies, and I have chosen to concentrate on fluorescence microscopy rather than other types of microscopy data—brightfield micrographs, electron micrographs, atomic-force microscopy, single-molecule localization microscopy, and others. We will briefly look at the analysis of gels and blots, but other cell biology data types (flow cytometry, proteomics, gene expression analyses, etc.) do not receive further attention. Analyses involving these types of data have many things in common with the approaches described in the book (i.e., experimental design, unbiased analysis, statistics, reproducibility, and presentation).

GOLDEN RULES

Throughout the book, you will find "golden rules" to follow. Here are the golden rules to being a digital cell biologist.

◆ ***Golden Rules***

- Quantification is key to understanding cell biology.
- Automate analyses wherever possible to minimize errors and bias introduced by humans.
- Aim for reproducible research to help the future you if no one else.
- Raw data are read-only.
- Outputs are disposable.