

# PMLS: Datasets

Here is a list of the datasets mentioned in the main text. Some are in plain text (human viewable), but the `.mat`, `.npy`, and `.npz` files are not. You can get them all in a single archive here:

<http://www.physics.upenn.edu/biophys/PMLS/Datasets/PMLSdata.zip>.

To get a file individually, try just clicking its link below (some browsers will then offer to save it to your hard drive). If your browser won't do that (for example, if a `.mat` file appears as a garbled page), try right-click (Windows) or ctrl-click (Mac), which should give you a context menu, including an item that allows you to download the file instead of attempting to display it.

Once you have the file, and if necessary have moved it to a convenient folder, you must next load it into your software. For MATLAB files:

- It may suffice to double-click the file in your computer's Finder.
- Or enter MATLAB and navigate to the folder containing the file, for example, by clicking the folder icon located on the upper left of the main MATLAB window, or using the `cd` command in the command window. Then you can double-click the file in MATLAB's **Current Directory** panel (upper left), or use the `load` command at the MATLAB command line or in a script. After this operation, your workspace will contain some variables with the data.
- Files with names ending in `.csv` or `.txt` are generic comma-separated-variable files. MATLAB can read such files by using the Import Wizard (**File>Import data**). But in most cases, the `.csv` file is just a duplicate of data also given in a `.mat` file. If you use MATLAB, and a `.mat` version exists, it's easier to use this file instead of the `.csv` or `.txt` version.
- Files with names ending in `.tif` are images, which can be read into MATLAB by using the `imread` command or Import Wizard.

For other software:

- Files with names ending in `.xls` or `.xlsx` are in Microsoft Excel format.
- Files with names ending in `.npy` or `.npz` are for Python users; the NumPy module can read them. Python can also read `.txt` and `.csv` files.

#1 = **HIVseries**: HIV infection time course. Files `HIVseries.mat`, `HIVseries.csv`, `HIVseries.npy` contain variable "a" with two columns of data. The first is the time in days since administration of a treatment to an HIV positive patient; the second contains the concentration of virus in that patient's blood in arbitrary units. (Data from A. Perelson. Modelling viral and immune system dynamics. *Nature Revs. Immunol.* (2002) vol. 2 (1) pp. 28–36 (Box 1).)

#2 = **population**: Files `population.mat`, `population.csv`, `population.npy`: First column: date in years CE. Second column: Estimated world population. (Data from

<http://www.vaughns-1-pagers.com/history/world-population-growth.htm>; see also

<http://www.census.gov/population/international/data/idb/worldpopinfo.php>.)

#3 = **shotNoise**: Files `shotNoise.mat` and `shotNoise.npy`: Variable A gives the arrival times of 290 photon absorption events in an avalanche photodiode detector. Time is measured in units of 50 ns. Total duration is 5 s. (Data courtesy John F Beausang.)

`shotNoise.txt`: Same data.

#4 = **brownian**: Jean Perrin's data on Brownian motion. Files `g26perrindata.mat`, `g26perrindata.txt`, `g26perrindata.csv`, `g26perrindata.npy` (data from J. Perrin, *Les Atomes*). The columns give  $x, y$  coordinates of the points in Figure 3.3a, in  $\mu\text{m}$ .

#5 = **LDexpt**: Luria-Delbrück experiment.

Files `LDexpt.mat`, `LDexpt.npz`: The variable `expcounts23` gives data for the experiment shown in Figures 4.6 and 4.8. Each culture studied had a certain number  $m$  of resistant bacteria, out of about  $2.4 \cdot 10^8$  bacteria total. The data specify a histogram of  $m$  with variable-width bins: Bin # $i$  includes outcomes for  $m$  in the range `bins(i)` through `bins(i+1)-1`. The number of cultures with  $m$  in this range is `expcounts23(i)`.

Another experiment, called #22 (not shown in the text), was similar. However, in this experiment, each culture was

sampled: only 1/4 of the total culture was withdrawn, spread on a plate, and checked for mutants. The result are in `expcounts22`, with the same bin structure as in experiment #23. This experiment had about  $2.8 \cdot 10^8$  bacteria per culture.

Files `LDexpt23.csv`, `LDexpt22.csv`, `LDexpt23.npy`, `LDexpt22.npy`: Same data with bins in the first column and `expcounts` in the second column.

(Data from Luria and Delbrück. Mutations of bacteria from virus sensitivity to virus resistance. Genetics (1943) vol. 28 pp. 491–511, table 5 p505.)

#6 = `clusters`: Geographic locations of a set of incidents. File `incidents.mat` contains the variables:

`incidents` =  $xy$  pairs describing map coordinates, arbitrary units, of each incident

`referencepoints` =  $xy$  pairs with map coordinates of four reference points

`place1,...` = names of reference points

Approximate scale is 44 m per a.u.

Files `incidents.csv`, `incidents.npy`, `reference.csv`, and `reference.npy` contain the same data. (File `clusters.npz` contains both `incidents` and `reference`.)

(Data from <http://bombsight.org/#14/51.4465/-0.0756>.)

#7 = `stocks`: Files `djiweekly.mat` and `djiweekly.npy` contain variable `djiweekly` with weekly closing values of the Dow Jones Industrial Average during the years from 1934 to 2014. File `djiweekly.csv` contains the same data in column 1.

#8 = `photodiodeblips`: Files `g112APDtraces.csv` and `g112APDtraces.npy` contain columns of data. Columns 1–2 correspond to higher illumination; 4–5 are medium illumination; 7–8 are for the lowest illumination. In each pair of columns, the first entry is time in seconds; the second is detector output in volts at that time. (Data courtesy John F Beausang.)

#9 = `myosinV`: Files `g42myosinwalk.mat` and `g42myosinwalk.npz`: The files contain these variables:

`yildizHistoRed` = stepping histogram for myosins taking only 70 nm steps;

`yildizHistoGreen` = stepping histogram for myosins taking  $(70 - x)$  nm steps alternating with  $x$  nm steps.

In each case, each row consists of the pair ((center of histogram bar in seconds), (observed frequency)).

`redDelta=0.5`, `greenDelta=1.0` are the respective bin widths in seconds.

Files `yildizHistoRed.csv`, `yildizHistoGreen.csv`, `yildizHistoRed.npy`, and `yildizHistoGreen.npy` contain the same information as the corresponding variables in the `.mat` file.

(Data from Yildiz et al. Myosin V walks hand-over-hand: Single fluorophore imaging with 1.5-nm localization. Science (2003) vol. 300 (5628) pp. 2061–2065, Fig. 6.)

#10 = `linearFitPoisson`: `linearFitPoisson.mat`, `linearFitPoisson.csv`, and `linearFitPoisson.npy` contain variables: `xvals`=(distance from radioactive source to detector, m)<sup>-2</sup>; `counts`=(counts in detector in a fixed time interval).

#11 = `vesicle`: `g293vesicle.mat`: The variable `amplitudes` is a list of bin centers; `frequencies` gives the corresponding number of times at which amplitudes falling in each bin were observed. The first bin contains the trials in which a stimulus evoked no response (failures). `g293vesicle.xlsx`, `g293vesicle.csv` and `g293vesicle.npy`: Same data. The first column contains amplitudes. The second column contains frequencies.

(Data from Boyd and Martin. The end-plate potential in mammalian muscle. J Physiol (Lond) (1956) vol. 132 (1) pp. 74–91, Fig. 8.)

#12 = `bursting`: `GoldingData.mat` and `GoldingData.npz`: Arrays `panelA1,2,3` contain data from three different trials. First column: time after induction, minutes Second column: sample mean of the number of mRNA molecules per bacterium.

Array `panelC`: First column:  $\log_{10}$  of the sample mean of mRNA count in steady state, for various degrees of induction. Second column:  $\log_{10}$  of the corresponding variances of mRNA count.

Arrays `panelD1,2,3` contain data from three different trials. First column: time after induction, minutes Second column: natural logarithm of the fraction of cells with zero copies of mRNA.

Files `panelA1.txt`, `panelA1.npy`, `panelA2.txt`, `panelA2.npy`, `panelA3.txt`, `panelA3.npy`, `panelC.txt`, `panelC.npy`, `panelD1.txt`, `panelD1.npy`, `panelD2.txt`, `panelD2.npy`, `panelD3.txt`, `panelD3.npy`: Same data.

(Data from Golding et al. Real-time kinetics of gene activity in individual bacteria. Cell (2005) vol. 123 (6) pp. 1025–1036 (see Fig. 8.6).)

#13 = HbMb: Binding curves of two macromolecules for oxygen. Files `hemoglobinmyoglobin.mat`, `hemoglobinmyoglobin.npz`, `hemoglobin.csv`, `hemoglobin.npy`, `hemoglobin.txt`, `myoglobin.csv`, `myoglobin.npy`, `myoglobin.txt`: variable `hemoglobin`: first column, concentration in M; second column, probability to be bound. variable `myoglobin`: same format.

(Data from F C Mills and M L Johnson and G K Ackers. 1976 Oxygenation-linked subunit interactions in human hemoglobin. Biochemistry 15:5350–5362 and A Rossi-Fanelli and E Antonini. 1958. Studies on the oxygen and carbon monoxide equilibria of human myoglobin. Arch Biochem Biophys 77:478–492.)

#14 = autoregulation: `rosenfeld.mat` and `rosenfeld.npz`: Synthetic governor. Variables named `regulatedX` are experimental data for regulated gene; `unregulated` is data for the unregulated system. In each variable, column 1 is time in units of cell cycles; column 2 is protein per cell normalized to its final value.

`rosenfeld.csv`, `rosenfeld.xlsx`: same data.

(Data from Rosenfeld et al. Negative autoregulation speeds the response times of transcription networks. J Mol Biol (2002) vol. 323 (5) pp. 785–793, Fig. 3.)

#15 = novick: Novick/Weiner data. `g149novickA.mat`, `g149novickA.txt`, `g149novickA.npy`: From their Fig. 1 (high inducer). First column: Time in hours. The e-folding time was about 3 hours. Second column: Fraction of maximum beta-galactosidase activity.

`g149novickB.mat`, `g149novickB.txt`, `g149novickB.npy`: From their Fig. 2. First column: Time in hours. The e-folding time was about 3 hours. Second column: Fraction of maximum beta-galactosidase activity.

(Data from Novick and Weiner. Enzyme induction as an all-or-none phenomenon. Proc Natl Acad Sci USA (1957) vol. 43 (7) pp. 553–566.)

#16 = catphoto: File `bwCat.tif`: A photograph of Emily. 864 × 648 pixels, 8-bit grayscale. You can import this to MATLAB by using

```
double(imread('bwCat.tif'))
```

Note the conversion needed to get the image from `uint8` type supplied by `imread`, to something you can do arithmetic on.

`bwCat.mat`: Same photo as a MATLAB array.

Files `gauss-filter.mat`, `gauss-filter.csv`, and `gauss-filter.npy`: These files contain a 45 × 45 array `gauss` specifying a Gaussian filter function.

#17 = stressFibers: `stressFibers.mat`, `stressFibers.npy`, `stressFibers.csv`, and `stressFibers.txt`: Image data of stress fibers in stem cells. (Data courtesy André Brown; see A Zemel, F Rehfeldt, A E X Brown, D E Discher, S A Safran. 2010. Nature Physics 6, 468–473; A Zemel et al. 2010. J. Phys.: Condens. Matter 22 194110.)