PtN: Datasets

Here is a list of the datasets mentioned in the book *From Photon to Neuron* (Princeton University Press 2017). 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/PtN/Student/

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 containing the data.
- Files with names ending in `.csv` or `.txt` are generic comma-separated-value 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 = `shotNoise`: Files `shotNoise.mat` and `shotNoise.npy`: Array `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 and Yale E Goldman.)
`shotNoise.txt`: Same data.

#2 = `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 and Yale E Goldman.)

#3 = `Mossbauer`: `rubyData.csv` contains the absorption spectrum of iron-57 exposed to gamma-ray light of various wavelengths. The first column gives the velocity of the source, a proxy for wavelength due to the Doppler shift. The third column gives count rate, proportional to the transmitted light intensity. (Data from Ruby and Bolef. Acoustically modulated γ rays from Fe^{57}. Phys. Rev. Lett. (1960) vol. 5 (1) pp. 5–7.)

#4 = `vesicle`: `g293vesicle.mat`: The array `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.)

#5 = `FRETdistance`: `FRETdistance.txt`: FRET efficiency as a function of distance; see Figure 2.22.
First column: donor-acceptor separation in units of 0.34`nm`. Second column: measured FRET efficiency.
(Data from Lee et al. Accurate FRET measurements within single diffusing biomolecules using alternating-laser excitation. Biophysical Journal (2005) vol. 88 (4) pp. 2939–2953.)

#6 = `emersonLewis`: Photosynthesis data; see Section 2.9.2. `emersonLewis.mat` and `emersonLewis.npz`: data from Figure 2.27a,b. These files contain arrays named `phycocyanin`, `carotenoids`, and `chlorophyll`: Column 1 gives wavelength in `nm`; column 2 gives the percentage of total absorbed light absorbed by each of the three pigment components at that wavelength. There is also an array named `QYield`: Column 1 gives wavelength in `nm`; column 2 gives the measured quantum yield of *Chroococcus* photosynthesis.
`phycocyanin.txt`, `carotenoids.txt`, `chlorophyll.txt`; `QYield.txt`;
`phycocyanin.xlsx`, `carotenoids.xlsx`, `chlorophyll.xlsx`; `QYield.xlsx`: same data in other formats.
(Data from Emerson and Lewis. The photosynthetic efficiency of phycocyanin in *Chroococcus,* and the problem of carotenoid participation in photosynthesis. J. Gen. Physiol. (1942) vol. 25 (4) pp. 579–595.)

#7 = `colorResponse`: File `responsecurves.mat` contains data shown in Figure 3.11 for relative spectral sensitivity of cone photoreceptor cells (*Macaca*, similar to human), corrected for absorption in the lens and macular pigment.
The file contains these arrays:
`lambdas(j)` = list of wavelengths in `nm` at which response is reported.
`sensitivity(i,j)` = relative sensitivities at wavelength `lambdas(i)`, where `j`=1 for L cones; 2 for M cones; and 3 for S cones.
File `responsecurves.csv`: Same data. First column are the entries in `lambdas`. Next three columns are entries in `sensitivity`.
(Data from D. Baylor, "Colour mechanisms of the eye" in *Colour: Art and science*, T. Lamb and J. Bourriau, eds. (Cambridge Univ. Press, Cambridge UK, 1995), Chapter 4, Fig. 6, which reprinted it from Baylor et al. Spectral sensitivity of cones of the monkey *Macaca fascicularis.* J. Physiol. (Lond.) (1987) vol. 390 pp. 145–160, fig. 3A.)

#8 = `superres`: File `simulateSuperres.npz` : Simulated camera data to illustrate localization microscopy. Contains the arrays:
`nPhotonList` = list of 6 numbers, the mean total photon count used for each of 6 simulated "exposures"
`nDataSets` = how many "video frames" were simulated for each "exposure"
`pixelCounts[i,j,k,m]` = photon counts in the pixel at `(i,j)`, in simulated video frame #`k`, with exposure #`m`.

#9 = `Yildiz`: File `YildizExpt.tif` : Experimental camera data courtesy Ahmet Yildiz, as a TIFF image stack (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). `YildizExpt.npz` : Same data, in a form convenient for Python programs to read. The file contains the array `imageStack[i,j,k]` containing light values in frame `i` at pixel coordinates `j,k`. These numbers represent a constant (non-random) camera offset, which we don't know in advance and must find, plus a random part. The random part is (true photon counts plus other dark noise)/4. We can model the other dark noise as Poisson with uniform intensity across all pixels.

#10 = `blinking`: `blinkData.mat`: Experimental data on the "blinking" of a single fluorescent molecule. Array `blinkOFF` gives the estimated probability density function of the durations of "off" states. Array `blinkON` gives the estimated probability density function of the durations of "on" states. In each array, the first column is $\log_{10}$ of the duration $t_w/(1\,\mathtt{s})$; the second column is $\log_{10}$ of the estimated probability density function $\wp_{on}(t_w)$, or $\wp_{off}(t_w)$, times $1\,\mathtt{s}$. The fluorophore was a tetraphenoxy-perylene diimide dye.
Files `blinkOFF.csv`, `blinkON.csv`: same data.
(Data from J. P. Hoogenboom, J. Hernando, E. M. P. H. Van Dijk, N. F. van Hulst, and M. F. Garcia-Parajo; see Hoogenboom et al. Accurate and unbiased estimation of power-law exponents from single-emitter blinking data. J. Chem. Phys. (2006) vol. 125 article 204713 pp. 1–12.)

#11 = `sakitt`: Files sakitt.mat , sakitt.xls: Sakitt experiment data. The array `Nsee(i,j,k)` gives the number of times that stimulus i ($1 \le i \le 3$) elicited rating $j-1$ ($0 \le j \le 6$) from subject k ($1 \le k \le 3$). The array `photonsin(i)` gives the strength of stimulus #i in mean number of photons presented to the cornea. (Data from Sakitt. Counting every quantum. J Physiol (Lond) (1972) vol. 223 (1) pp. 131–150, Table 1.)

#12 = `ProbSeeRodCell`: File BaylorNunnSchnapf.mat: Baylor et al. data, shown as the points in Figure 9.9. The data give the probability of seeing for macaque rod cells as a function of the density of photons supplied to the rod outer segment.
The array `log10Nbar` contains the logarithm of the density of photon arrivals (units of photons per $\mu\mathsf{m}^2$, applied over $50\,\mu\mathsf{m}^2$). The four columns correspond to four different cells; the five rows correspond to five different flash intensities.
The array `Psee` contains the corresponding fractions of 65 trials that elicited a rod response.
File BaylorNunnSchnapf.csv: Same data with the flash intensities in the first four columns as above, and `Psee` in the next four columns. (Data from Baylor et al. The photocurrent, noise and spectral sensitivity of rods of the monkey Macaca fascicularis. J Physiol (Lond) (1984) vol. 357 pp. 575–607, Fig. 8.)

#13 = `currentRodCell`: g109currentRodCell.mat: The array `histoA` contains data shown in Figure 9.12a. The array `histoB` contains data shown in Figure 9.12b.
The array `histoA`: Responses to dim flashes. First column is a list of bin centers for peak current; second column gives the corresponding number of times at which amplitudes falling in each bin were observed. The array `histoB`: Responses to stronger flashes.
g109currentRodCell.xlsx and g109currentRodCell.csv: same information.
(Data from F Rieke, "Seeing in the dark: Retinal processing and absolute visual threshold," in *The senses: A comprehensive reference,* R H Masland and T Albright, eds. (Academic Press, San Diego CA 2008), volume 1, pp. 393–412.)

#14 = `onePhotonRod`: File g313onePhotonRod.mat and g313onePhotonRod.npz: The arrays give binned data for rod response:
`currNoPhot`, `pNoPhot`: current values and corresponding estimated probabilities for responses to zero photons.
`currOnePhot`, `nOnePhot`: current values and corresponding estimated probabilities for responses to exactly one photon.
currNoPhot.csv, pNoPhot.csv, currOnePhot.csv, pOnePhot.csv: same information.
(Data from Phan et al. Physical Review Letters (2014) vol. 112 (21) pp. 213601.)

#15 = `gatedChannels`: File gatedChannels.mat: Dose-response relation for opening of cyclic nucleotide-gated channels by cGMP (Figure 10.9). The arrays in this file are:
`cGMP`: concentration of cGMP [μM], `curr`: ionic current [a.u.]; `currBot` and `currTop`: current, bottom and top of each error bar.
gatedChannels.csv: same information.
(Data from Nakatani and Yau. Guanosine 3′,5′-cyclic monophosphate-activated conductance studied in a truncated rod outer segment of the toad. J. Physiol. (Lond.) (1988) vol. 395 pp. 731–753, Fig. 8d.)

#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.
squareRect.tif: A set of geometrical objects, which may show more clearly the effects of different kinds of filter.
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 = `17Planck`: File wavelengths.csv : A list of wavelengths (`nm`) at which spectral irradiance was measured for two hot objects (actually the same object, a light-bulb filament, at two different unknown temperatures). File

planck.csv : The first column is spectral irradiance in arbitrary units when the bulb is run at 73 W; the second is for 55 W (same arbitrary units as the other case).