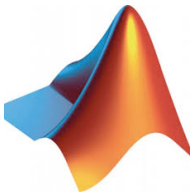
M2D7:
Assess protein function

11/03/2015

note: no class nor lab on 11/10 - 11/11 ☺

other note: you can no longer switch Journal Club papers



Today in lab

- Quiz 4
- Lab notebook graded: M2D3 andreakw@
- Analyze data with Excel
- Analyze data with MATLAB
- Ask questions
  - about M2D7
  - about all of M2

A few more assignments for M2

- Methods (due M2D6)
  - you did really well!
  - handed back (with peer's feedback) on M2D8
- Protein engineering research report
  - due by 5pm on Sunday, November 15
  - extra office hours, come with specific questions

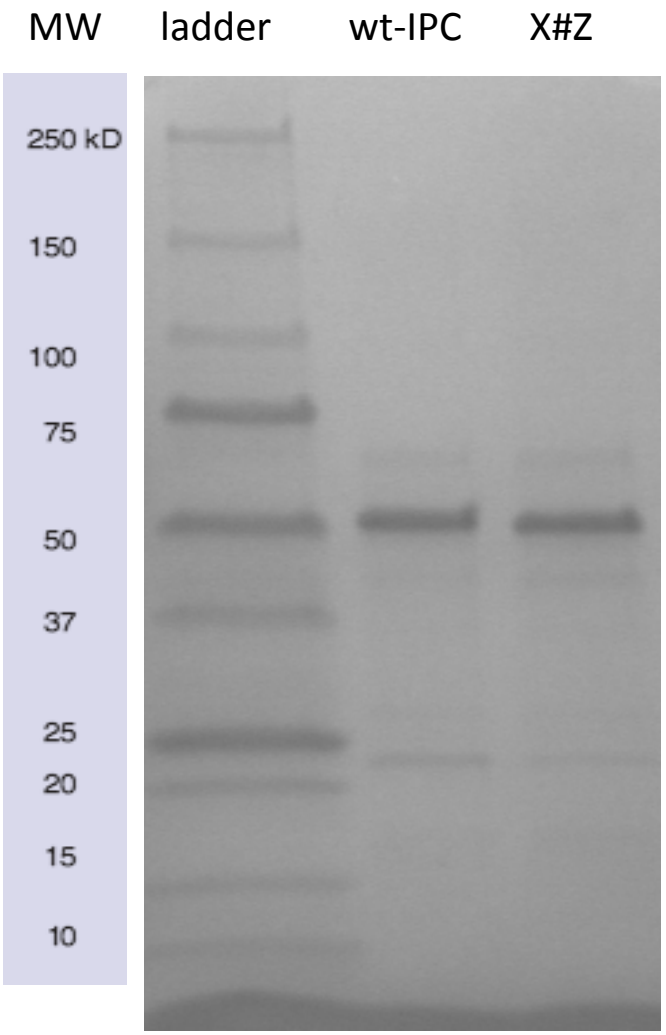
    to 56-302 on Wed., Nov. 11th, 10am-4pm

    to 16-239 on Thu. and Fri., Nov. 12th-13th, 10-11am

    to 16-317 on Thu. and Fri., Nov. 12th-13th, 6-8pm

    to 16-429b on Fri., Nov. 13th, 10am-12pm
- Blog post
  - due by 8pm on Sunday, November 15
  - write about M2, journal club, etc…

Estimate protein concentration using Bradford assay or unstained ladder

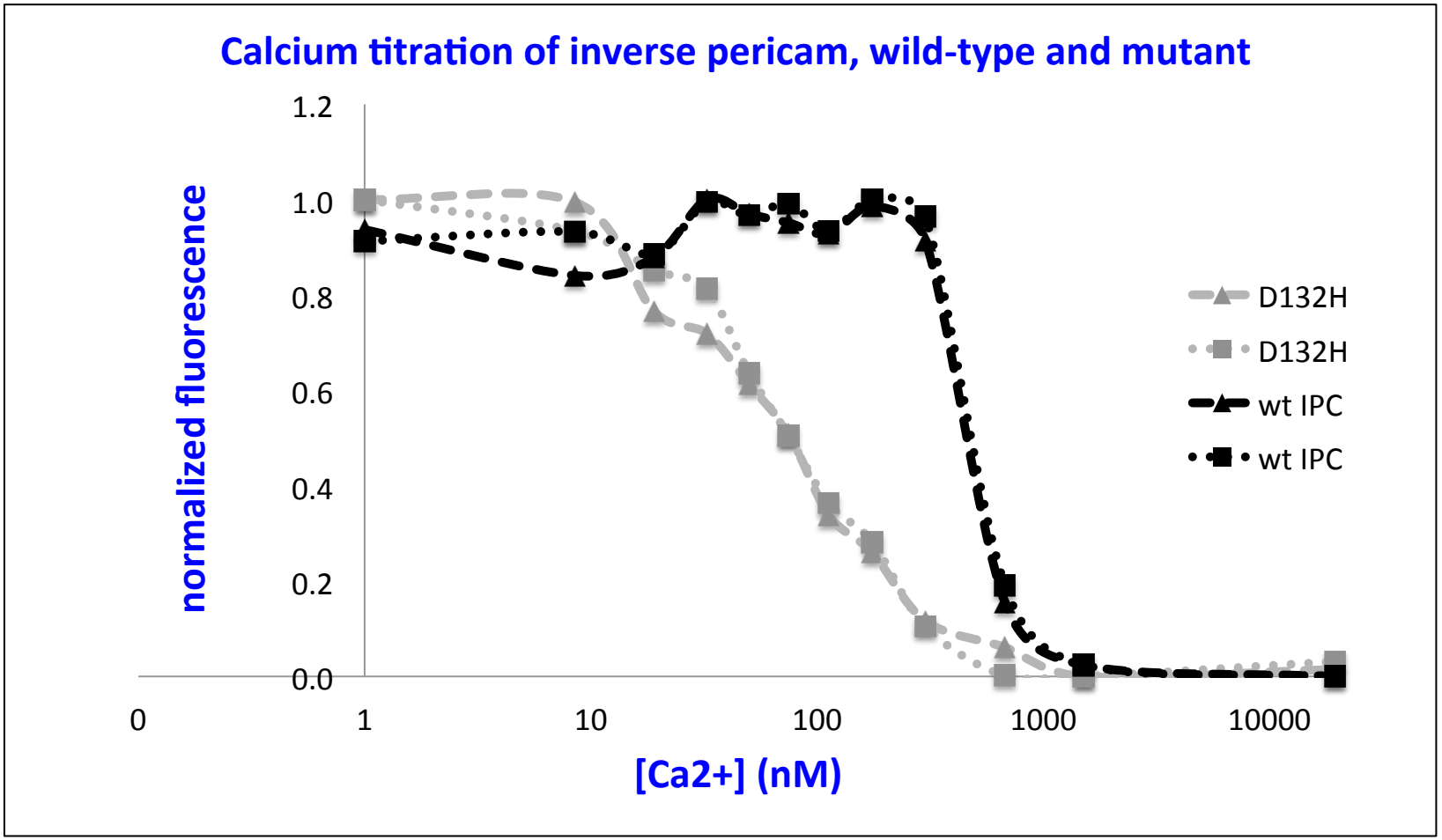


- Bradford assay extrapolation from BSA calibration curve
  - don't forget units!
  - mg/mL
- Or comparison with unstained ladder
  - 50 kDa band has 7.5 ng / 10 µL
  - your wt-IPC is 2 x as bright
  - hence [wt-IPC] = 15 ng / 18 µL
  - and [IPC(X#Z)] = ~15 ng / 18 µL
  - Convert to M (mol/L) using 1 Da = 1 g/mol

Plot your IPC-calcium titration data in Excel

- Normalize data (or average of 2 data sets):

$$S = \frac{F - F_{\min}}{F_{\max} - F_{\min}}$$



Analyze data further in MATLAB

1. Enter your data:
   - L = [ligand] = [Ca^{2+}] in µM
   - S_wt: signal wild-type IPC
   - m1 is *your* mutant, m2 is another team's

normalized fluorescence signal (from Excel)
12 values
if 1 --> enter 0.999
if 0 --> enter 0.001

2. `logspace` (a, b, N)
   - generates a row vector of N logarithmically equally spaced points between decades 10^a and 10^b.
   - choose a = -3 , b = 2 , and N = 10,000

(from 1 nM to 100 uM)

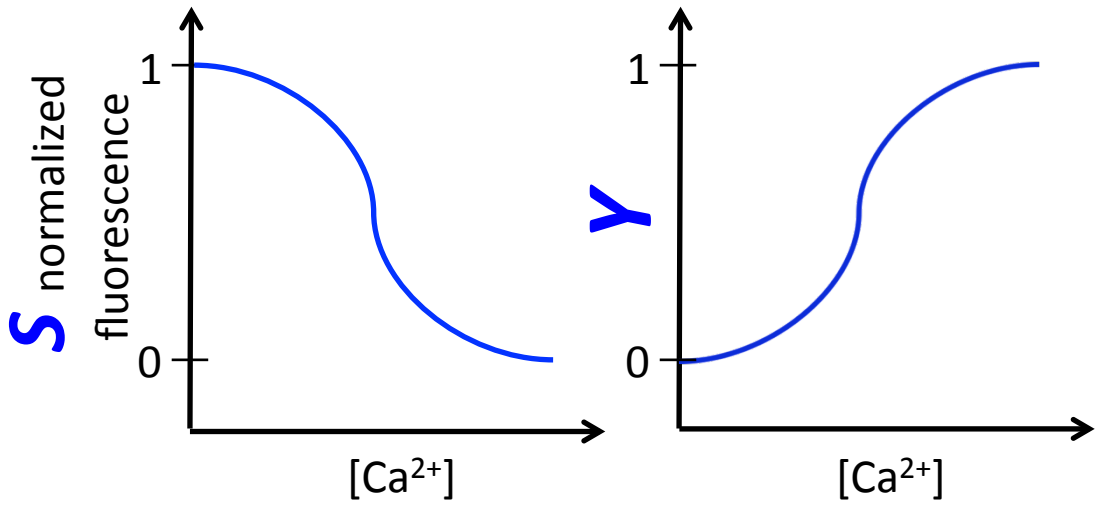
3. `A ./ B`
   - divides element by element

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 6 & 9 \\ 4 & 8 & 12 \end{bmatrix} ./ \begin{bmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}$$

MATLAB code analyzes data along 3 models

- As in lecture, convenient to go back to *pericam* formalism in equations:

$$Y = 1 - S$$



Part 1: fit apparent K_d

$$Y = \frac{L}{K_d + L}$$

Part 2: fit K_d and n

$$Y = \frac{L^n}{K_d^n + L^n}$$

Part 3: fit K_d and n

Hill analysis

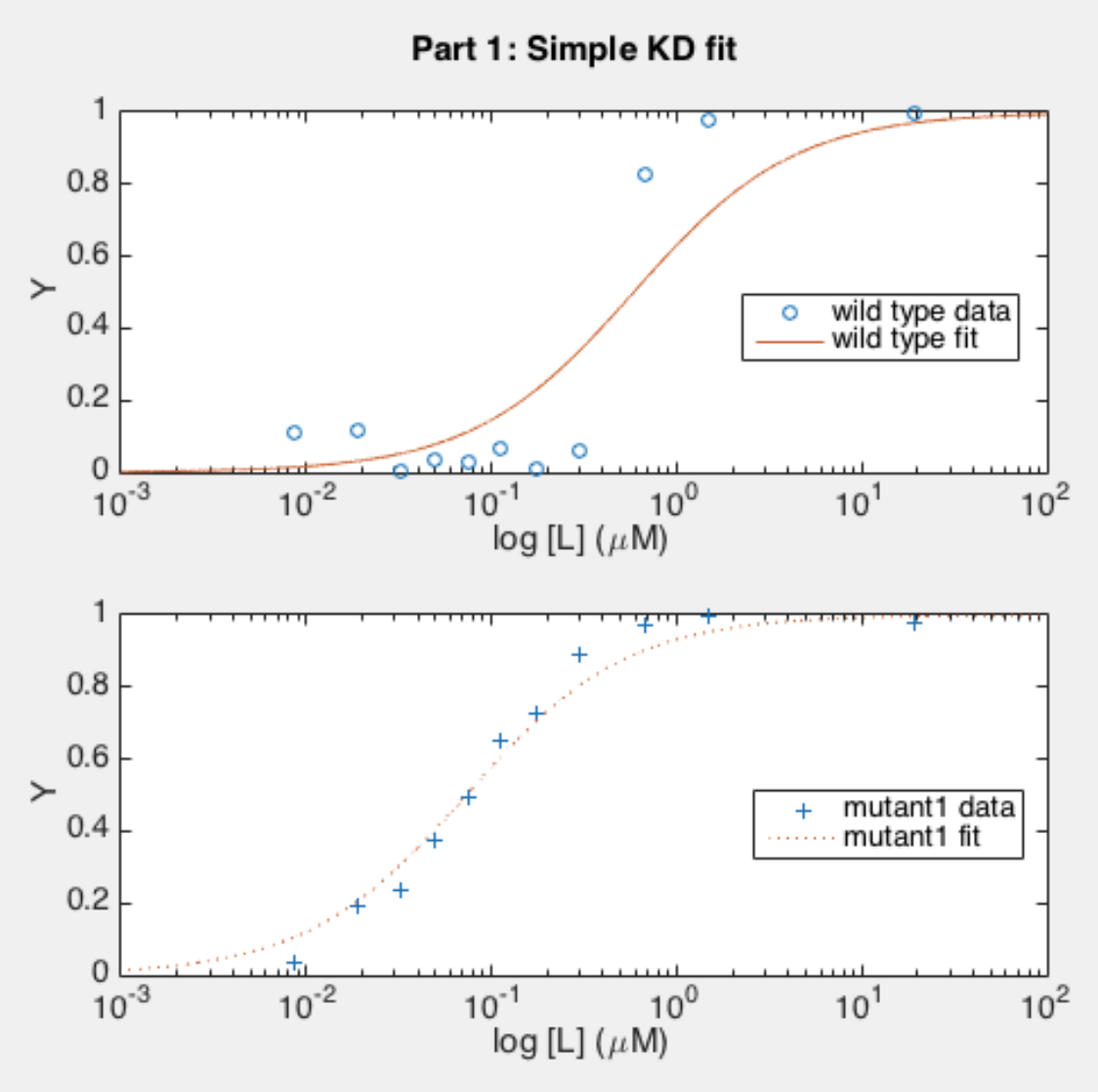
$$\log\left(\frac{Y}{1-Y}\right) = n\log(L) - n\log(K_d)$$

Would anyone like to go through the derivation of

$$Y = \frac{L}{K_d + L}$$?

Part 1: fit apparent K_d

$$Y = \frac{L}{K_d + L}$$

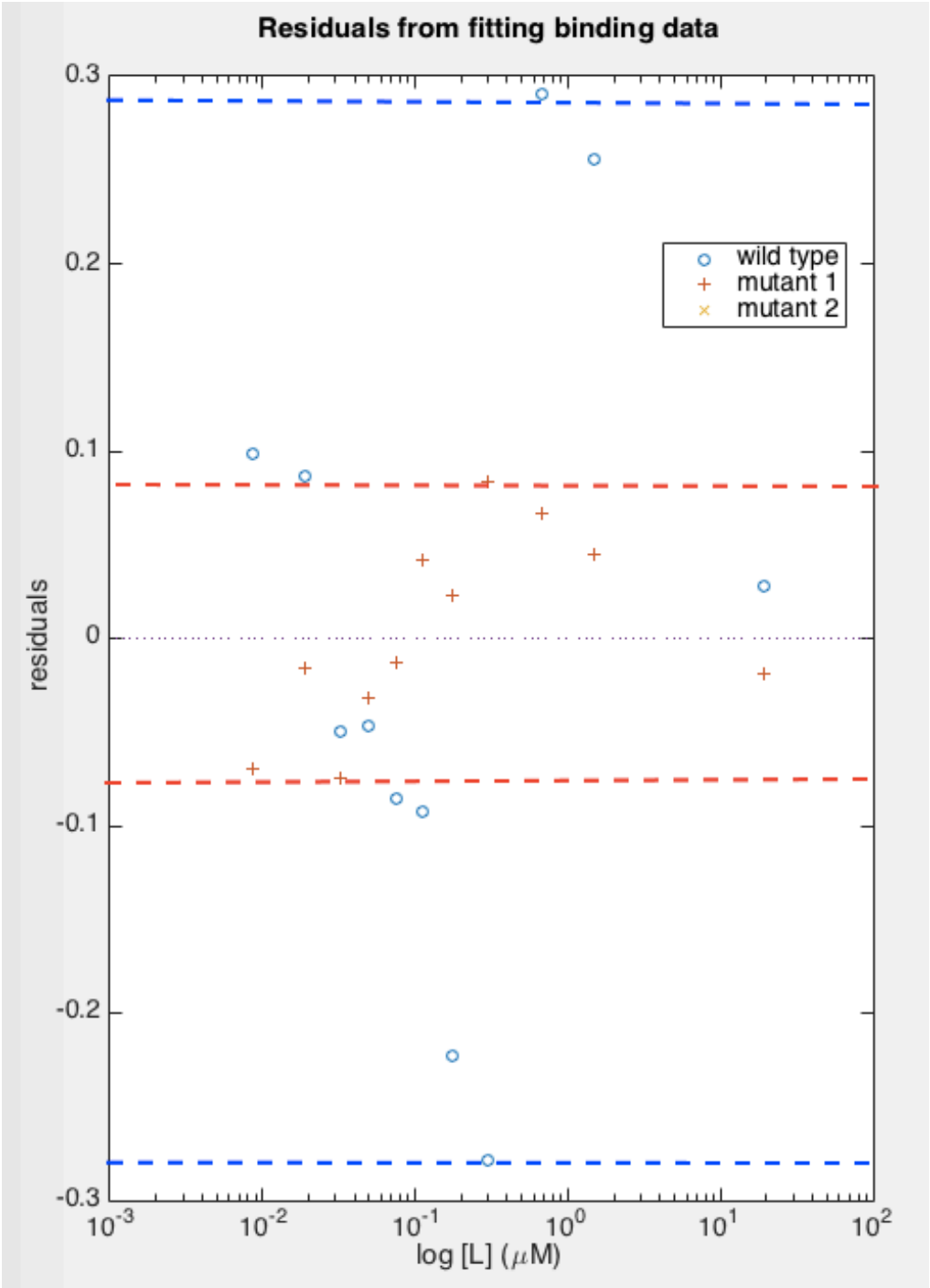



```
KD1_wt = 0.5858 μM

KD1_m1 = 0.0729 μM
```

- How good is the fit?
  - for wt-IPC?
  - for mutant?

confirmation that the mutant has a higher affinity to calcium than the wild-type IPC.

poor fit, in particular to describe wt-IPC.



Part 1

$$Y = \frac{L}{K_d + L}$$

- How good is the fit?
  - for wt-IPC?
  - for mutant?

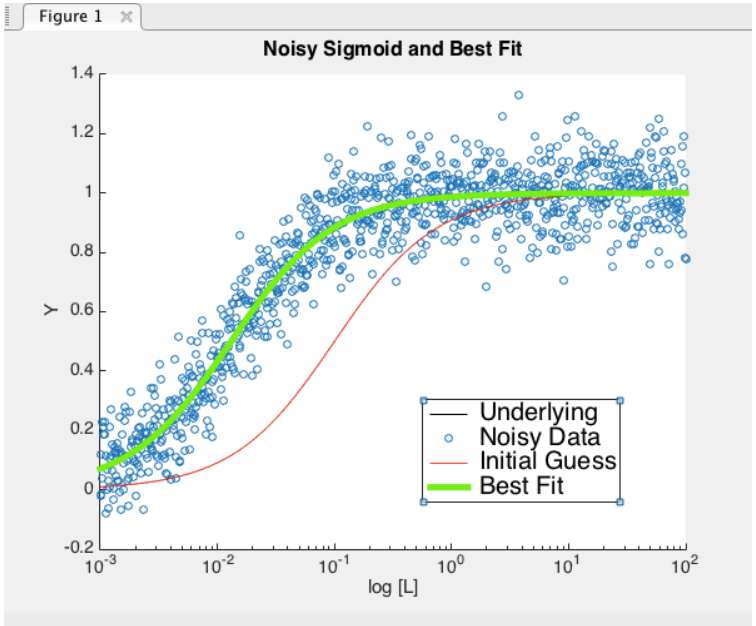
➢ Quantify *residuals*: distribution and amplitude

residuals = Yexperimental - Ymodel

Nonlinear regression is at the core of the MATLAB code

- `nlinfit( X, Y, @model, initialGuess )`
  - X (predictors): calcium concentrations
  - Y (responses): fluorescence signal
  - model: `Fit_SingleKD`

    ```
    x ./ ( KD + x );
    ```
  - initialGuess: starting value for `KD` **can be a vector: [KD0, n0]**



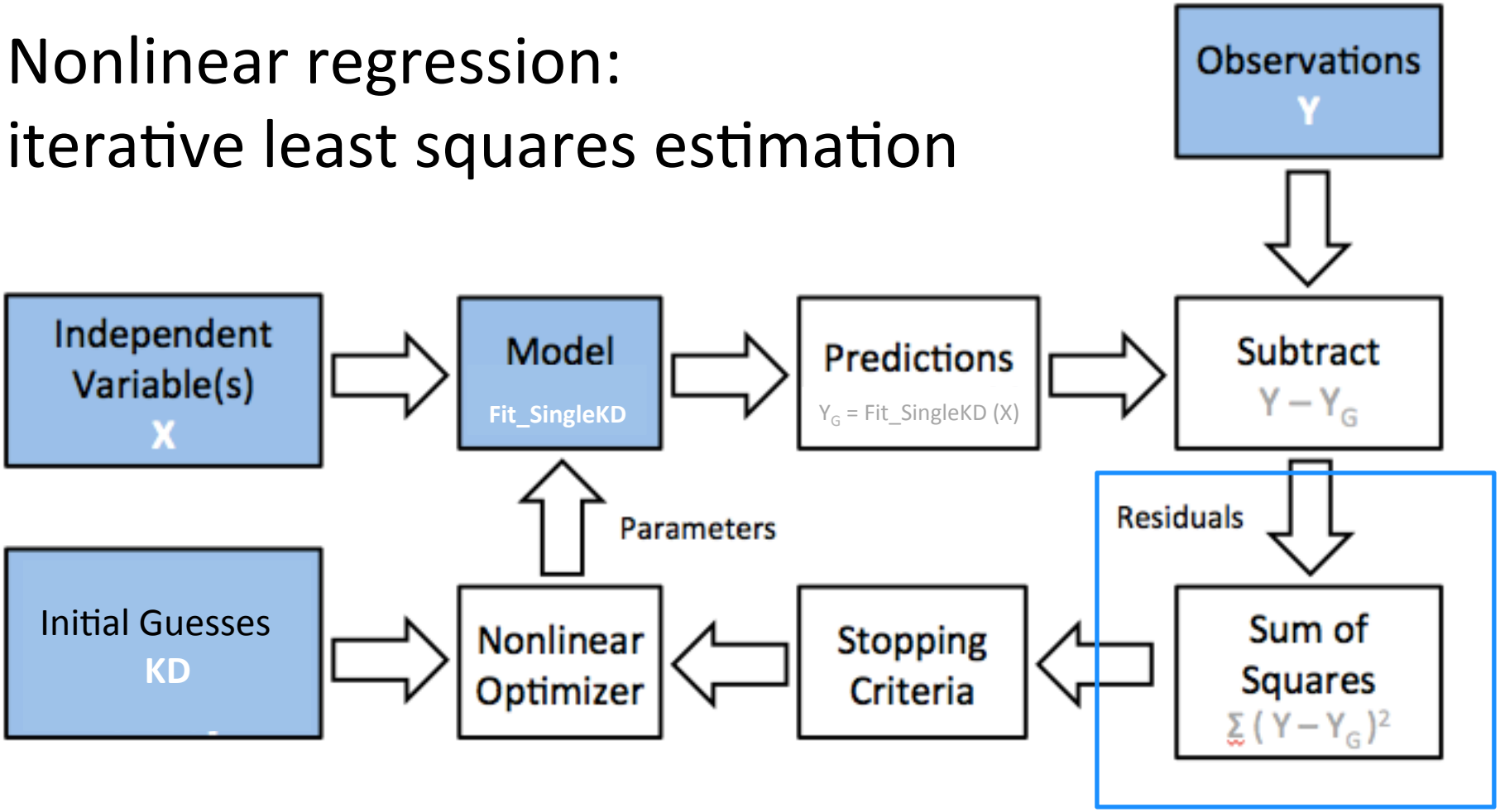
- Find parameters that can explain

  `Y = model( Parameters, X )`

  and start your search with

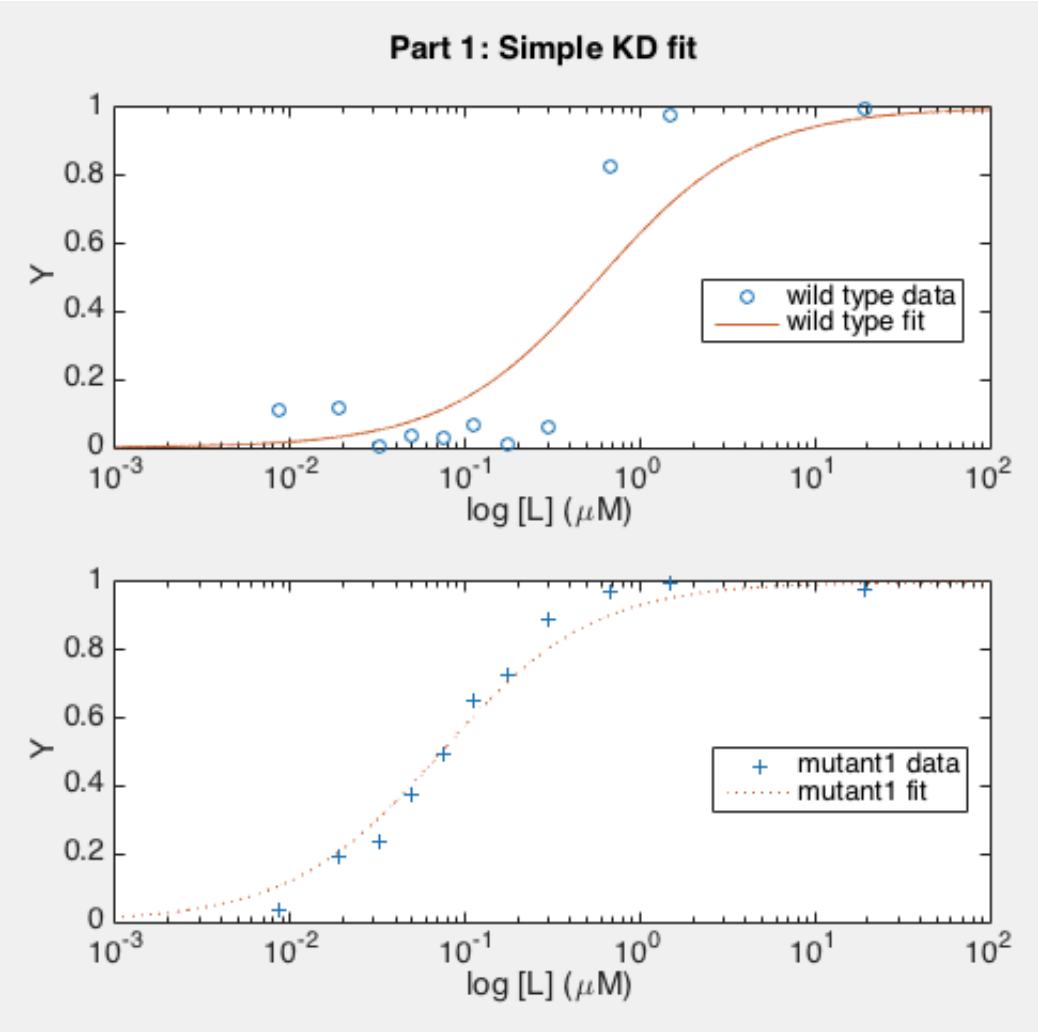
  `parameters0 = initialGuess`

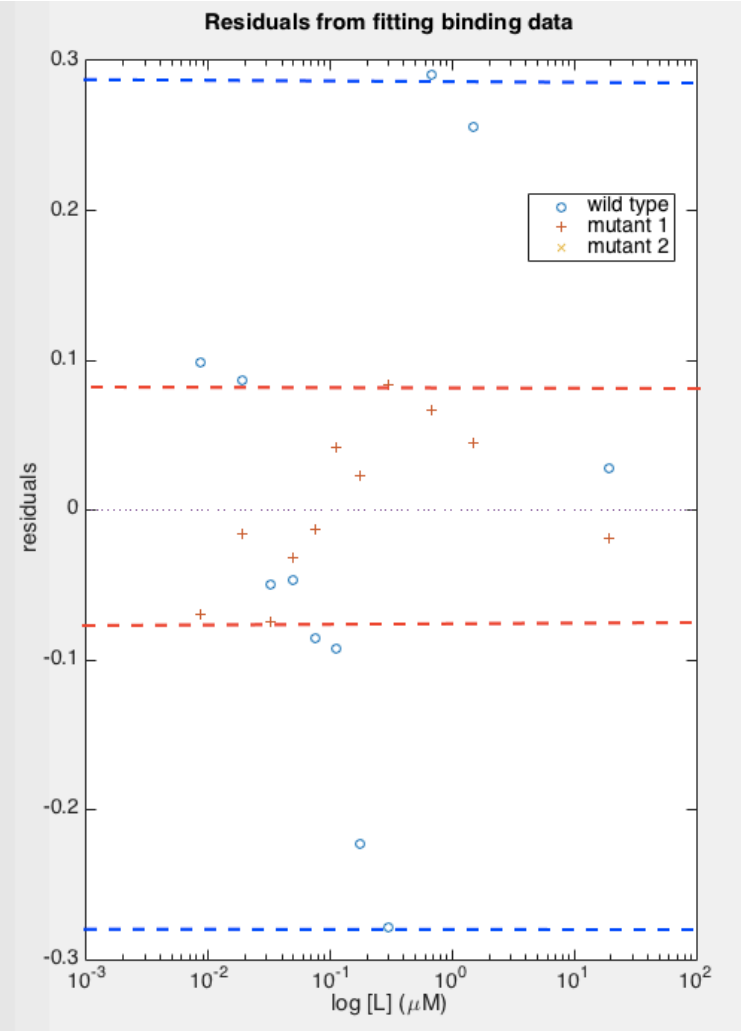
Nonlinear regression: iterative least squares estimation



- Optimum reached = changing any of the parameters will result in a higher residual sum of squares.
- Optimizer stops when parameters or sum of squared residuals changes less than tolerance, or when maximum number of iterations reached.

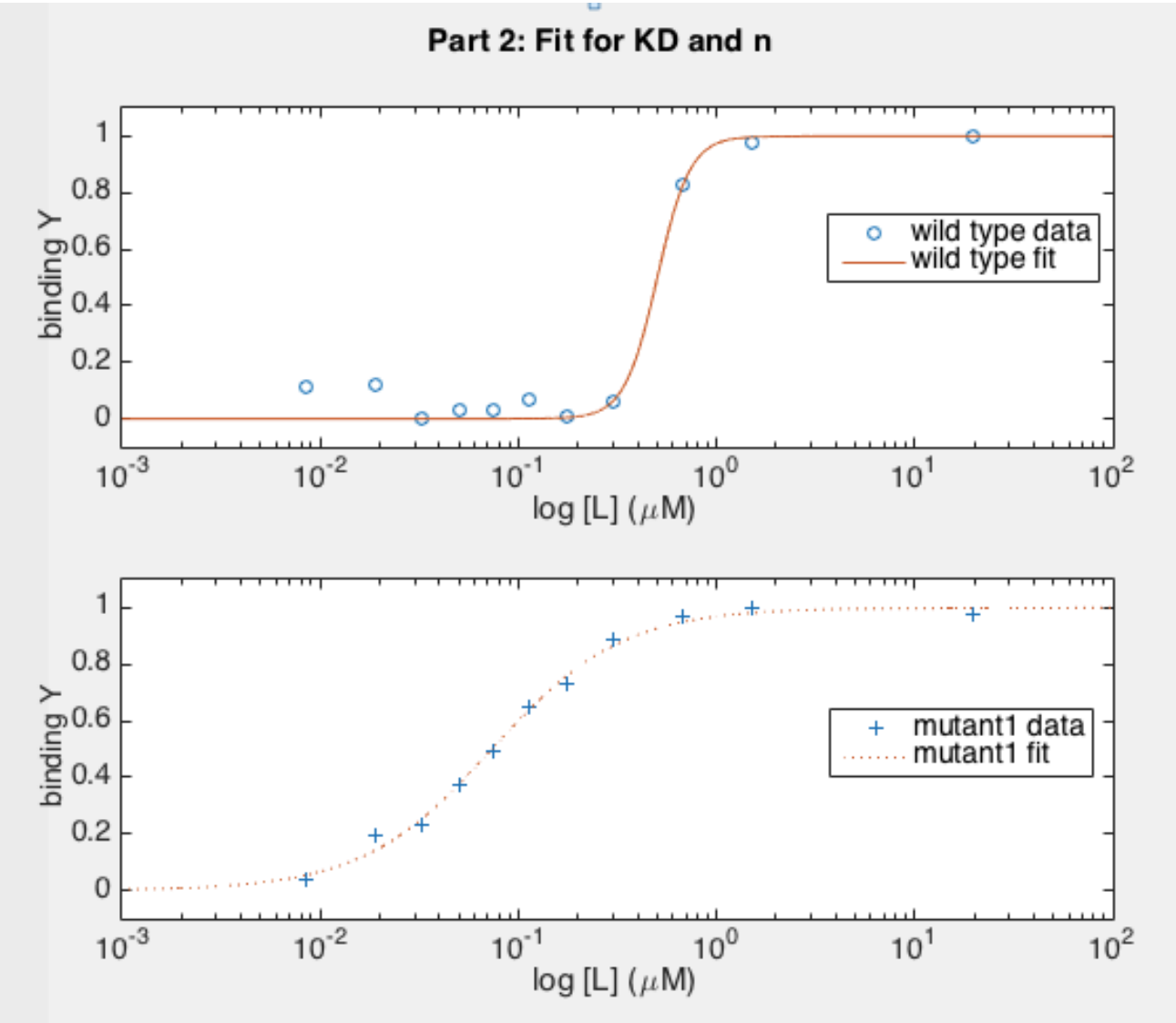
… and this is why residuals $Y - Y_{model}$ provide qualitative and quantitative goodness of fit!





Part 2: fit K_d and n

$$Y = \frac{L^n}{K_d^n + L^n}$$



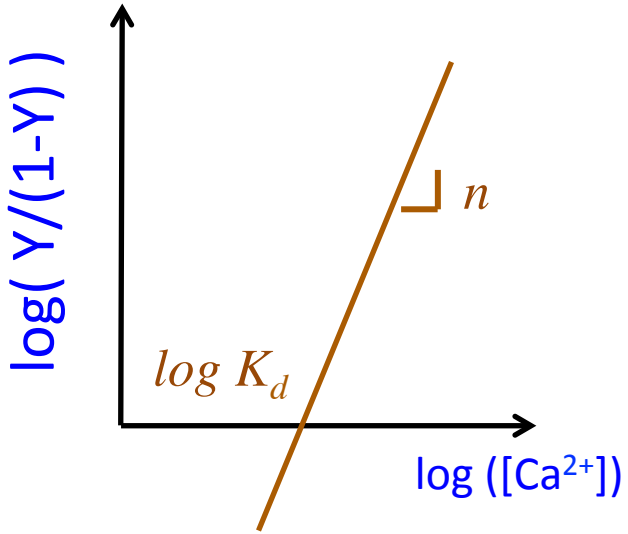

```
KD2_wt = 0.5025 μM
n_wt = 5.2508

KD2_m1 = 0.0737 μM
n_m1 = 1.3250
```

- Is the fit any better?

Part 3: fit K_d and n by Hill analysis

$$\log\left(\frac{Y}{1-Y}\right) = n\log(L) - n\log(K_d)$$

- Work only on linear transition region
  - linear fit (polynomial of degree 1)
  - x-intercept = log (K_d)
  - slope = n
- Will need to change indexes in MATLAB algorithm
  - then work with *cell arrays* to parallelize analysis




```
L_wt = L(9:10); Y_wt = Y_wt(9:10); Yp_wt = Y_wt./(1-Y_wt);
L_m1 = L(2:10); Y_m1 = Y_m1(2:10); Yp_m1 = Y_m1./(1-Y_m1);
L_m2 = L(6:10); Y_m2 = Y_m2(6:10); Yp_m2 = Y_m2./(1-Y_m2);

% Create cell arrays to concatenate elements of different size:
L = {L_wt; L_m1; L_m2};
Y = {Y_wt; Y_m1; Y_m2};
Yp = {Yp_wt; Yp_m1; Yp_m2};
```

Make a story out of your M2 results