Stochastic simulation of reaction networks

Week 5

Basic Outline: 3 functions

```
gilldriver

[t,dimer]=mygill(tfin,rtab,x,c)

h=update(x)
```

[within driver]

Gilldriver

What does the driver do?

[within driver]

Gilldriver

What does the driver do?

- Set parameters (these are given in notes)
- Set **rtab** (hard-code)
- Run Gillespie (**mygill**) **nr** times for first 6 reactions
  - Plot mean and STD
- Run Gillespie (**mygill**) **nr** times for all 8 reactions
  - Plot mean and STD

[within driver]

Reactant indices always in least to greatest order for those in *that reaction*

Building **rtab**

- General idea: alternate between index and #

{[reactant index #gain/lose reactant index #gain/lose]

[reactant index #gain/lose …. for however many reactants]

… [indices and gain/loss for last reaction]}

For example, take reactions:

A has index 1;
B has index 2;
C has index 3

A+B →C

A→2B

rtab={[1 -1 2 -1 3 1] (lose an A, lose a B, gain a C)

[1 -1 2 2]} (lose an A, gain 2 Bs)

Practice

Regarding the rtab convention established in the notes, if I wish to encode the reaction

$X_3 \text{ --> } X_1 + X_2 + 10X_4$,

then the associated row of rtab should be:

A. [1 2 3 4 1 1 -1 10]

B. [1 1 2 1 3 -1 4 10]

C. [1 -1 2 -2 3 1 4 -10]

[within driver]

Your turn

Work with the person next to you to build **rtab** by hand for the reactions for this project.

Indices	
R:	1
Pr:	2
RPr:	3
P:	4
D:	5
I:	6

$R + Pr \rightarrow RPr$

$RPr \rightarrow R + Pr$

$RPr \rightarrow 10P + R + Pr$

$P + P \rightarrow D$

$D \rightarrow P + P$

$P \rightarrow$

$Pr + D \rightarrow I$

$I \rightarrow Pr + D$

Remember at first we only want to run Gillespie on these 6 reactions. Then run again for the *whole* **ruletab**!

How can we represent the first 6 rows of ruletab for our first call to mygill?

After setting values and building ruletab, run mygill

- Run mygill nr times (value specified earlier in code)
- `[t,dimer]=mygill(tfin,rtab,x,c)`
- What is x? What is c? Where do we find those?
- Interpolate. But before addressing this, a word on *time*:
  - we want to plot dimer vs. t for all times from 0 to tfin (100), at an increment of tinc (.1).
  - Build tvec as specified in notes
  - Each of the *nr* times you run mygill, interpolate *t* and *x* with *tvec* (see next slide).

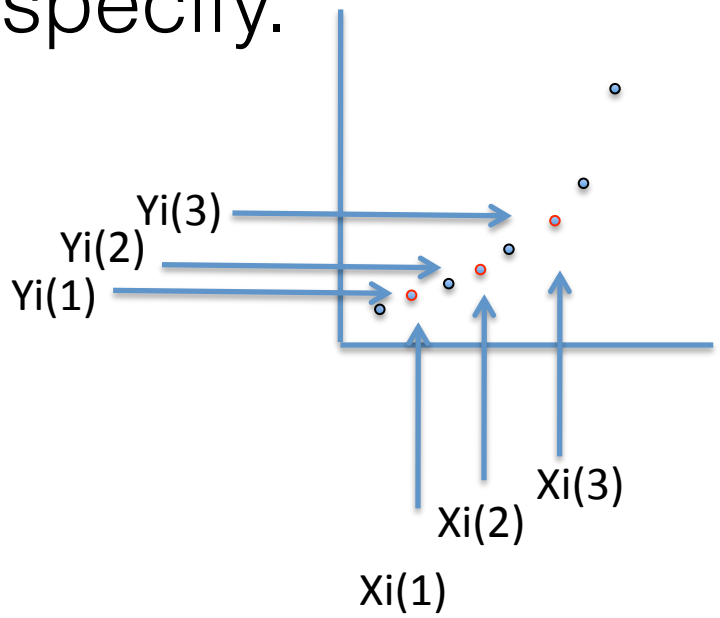
[within driver]

Hint: find STD using std(x) in MATLAB

Using interp1

Use: YI = INTERP1(X,Y,XI)

Basic interpolation: you have y(x): vector of y values for given vector of x values. You want to find vector of yi values for vector of xi values that you specify.



From notes (modify using interp):

```
for j=1:nr
[t,x] = mygill(tfin,rtab,x0,c);
 X(j,:) = x;
end
 avg = mean(X,1);
--find STD
```

h = update(x)

- h is a vector, each of whose elements represents the number of ways the corresponding reaction can occur.

  Remember that x contains the quantities of each reactant.

  R + Pr → RPr h(1) = # ways this reaction can occur
  RPr → R + Pr h(2) = # ways this reaction can occur
  RPr → 10P + R + Pr …
  P + P → D
  D → P + P
  P →
  Pr + D → I
  I → Pr + D h(8) = # ways this reaction can occur

Write down h(1) through h(8) as functions of elements in x.

[within mygill]

[t, dimer]=mygill(tfin,rtab,x,k)

"1. The user is unsure of how to choose maxiter and would prefer to provide the final time, tfin."

→ Substitute **while** for **for** (how?)

-first call update h=update(x)

-Cryptic notes: The use of update is relatively straightforward, e.g., a = c. * h; With this a in hand you may now use **cumsum** and **find** to find the next reaction index (in just a couple of lines - and with NO if clauses). With this index in hand you may visit the proper element of **rtab**.

-record your dimer count before cycling through while loop again. (Which element of x is dimer?)

[within mygill]

Within the while loop

see lotkaedit.m

- Update counter
- Call update function
- Reaction probability: $a_m = h_m c_m$
- "We shall have occasion to call on:"

  $a_0 = a_1 + a_2 + \cdots + a_M$ use MATLAB's sum function
- Generate random number r_1
- Set t(i)=t(i-1)+T (tau in lotka) (solve 8.1 for T); 8.1: $r_1 = \exp(-a_0 T)$
- Generate random number r_2
- use cumsum and find and subsequently access elements of rtab to update x (quantites of each reactant present)
- Store dimer count, x(5).