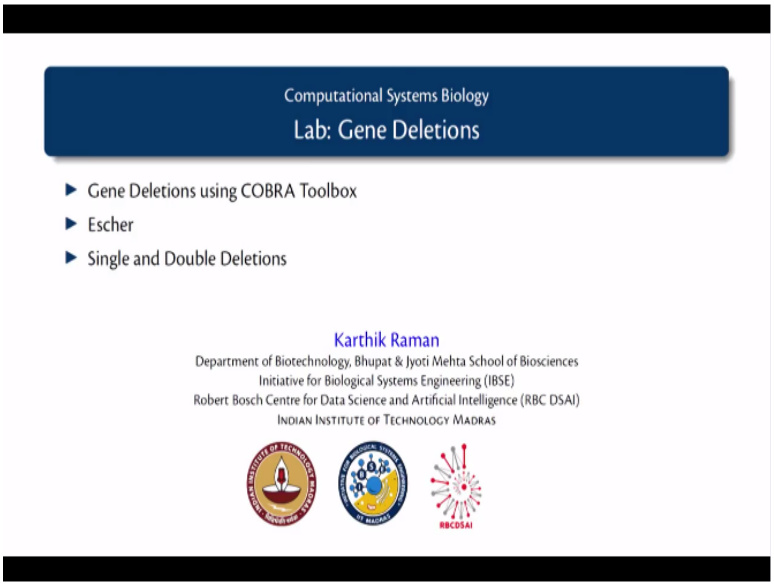
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology – Madras

Lecture - 71
Lab: Gene Deletions

(Refer Slide Time: 00:11)



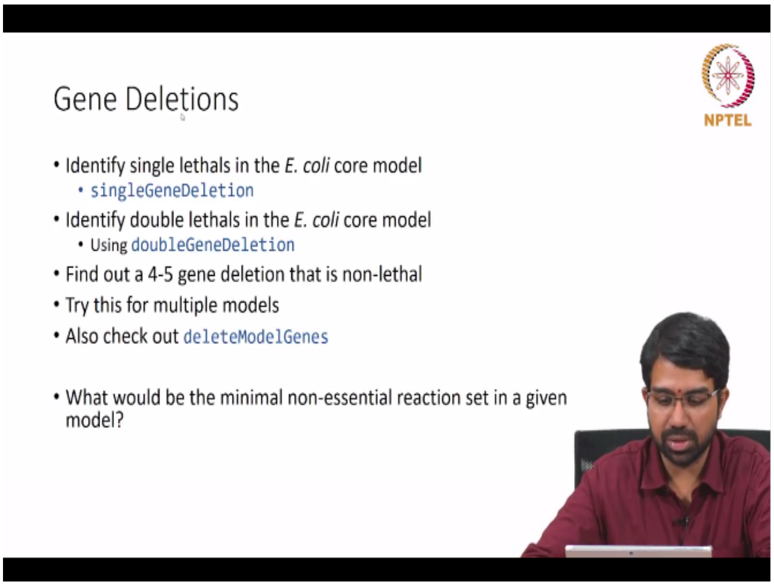
Computational Systems Biology
Lab: Gene Deletions

- Gene Deletions using COBRA Toolbox
- Escher
- Single and Double Deletions

Karthik Raman
Department of Biotechnology, Bhupat & Jyoti Mehta School of Biosciences
Initiative for Biological Systems Engineering (IBSE)
Robert Bosch Centre for Data Science and Artificial Intelligence (RBC DSAI)
Indian Institute of Technology Madras

In today's lab session, we will focus on gene deletions, how we perform gene deletions using the COBRA toolbox. I will also show you this tool called Escher which gives you a nice visual picture of what happens when you delete genes, how the rerouting of reactions happens and so on and we will also look at how we perform single or double or synthetic lethal deletions, identify synthetic lethals using the COBRA toolbox.

(Refer Slide Time: 00:33)



Gene Deletions

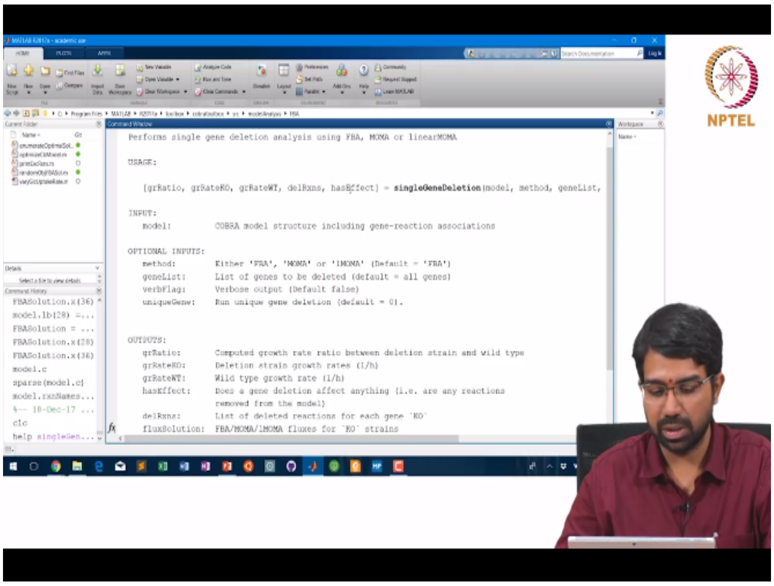
- Identify single lethals in the *E. coli* core model
  - `singleGeneDeletion`
- Identify double lethals in the *E. coli* core model
  - Using `doubleGeneDeletion`
- Find out a 4-5 gene deletion that is non-lethal
- Try this for multiple models
- Also check out `deleteModelGenes`
- What would be the minimal non-essential reaction set in a given model?

So welcome back. Let us try out the following problems today in the lab. Let us identify single lethals in the E. coli core model using the single gene deletion function in COBRA toolbox. Let us identify double lethals in the E. coli core model using the double gene deletion and slightly more tricky problem is can we find out a 4-5 gene deletion so remove 4 or 5 genes from E. coli without killing it right.

And I will show you there is an interesting website that can be used to visualize these as well and try this for multiple models not just the E. coli core model but for the latest E. coli model or mycobacterium tuberculosis and so on. These can be downloaded from the BIGG database, bigg.ucsd.edu, so for deleting genes it is good to check out the function delete model genes right.

And lastly what would be the minimal non-essential reaction set in a given model, how would you compute that right? So sort of minimal genome right, so if you remove any reaction the organism is not going to survive so how would you come up with such a reaction set. So these are some of the questions I want you to tackle today.

(Refer Slide Time: 01:46)

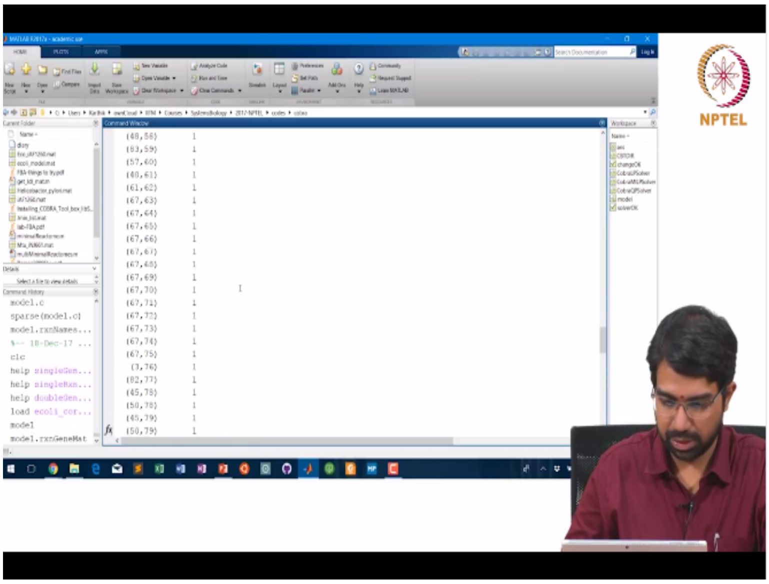


```
Performs single gene deletion analysis using FBA, MOMA or linearMOMA

USAGE:

   [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleGeneDeletion(model, method, geneList,

INPUT:
   model:         COBRA model structure including gene-reaction associations

OPTIONAL INPUTS:
   method:        Either 'FBA', 'MOMA' or 'lMOMA' (Default = 'FBA')
   geneList:      List of genes to be deleted (default = all genes)
   verbFlag:      Verbose output (Default false)
   uniqueGene:    Run unique gene deletion (default = 0).


OUTPUTS:
   grRatio:       Computed growth rate ratio between deletion strain and wild type
   grRateKO:      Deletion strain growth rates (1/h)
   grRateWT:      Wild type growth rate (1/h)
   hasEffect:     Does a gene deletion affect anything (i.e. are any reactions
                  removed from the model)
   delRxns:       List of deleted reactions for each gene `KO`
   fluxSolution:  FBA/MOMA/lMOMA fluxes for `KO` strains
```

So this is a single gene deletion function, the outputs are growth ratio, growth rate on knockout so this is basically growth rate or knock out by growth rate of wild type right gr ratio and what are the reactions that are released and deleted and if there is any effect for the deletion hasEffect and you also have a help single reaction deletion which is again very similar.

You also have a double gene deletion right. This essentially gives you a sort of matrix. **"Professor - student conversation starts."** Yes, the model has that information. **"Professor - student conversation ends."**

(Refer Slide Time: 02:56)



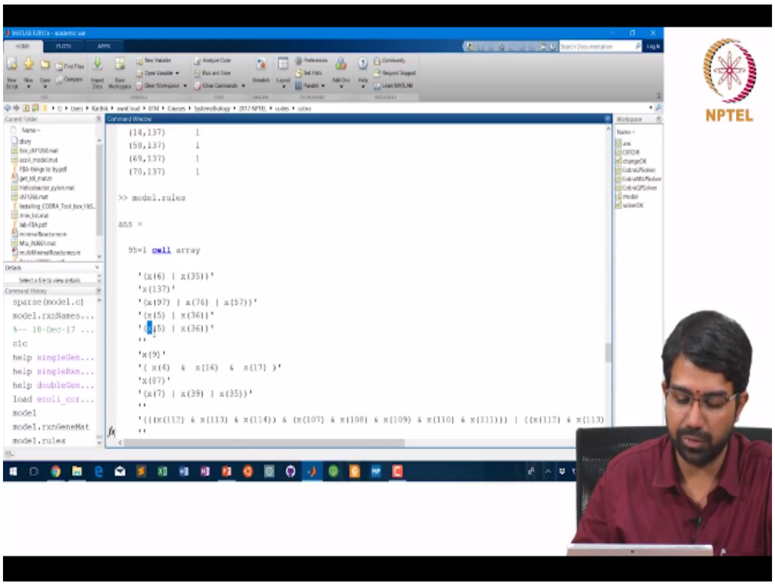
```
model.c
sparse(model.c)
model.rxnNames...
%-- 18-Dec-17 ...
clc
help singleGen...
help singleRxn...
help doubleGen...
load ecoli_cor...
model
model.rxnGeneMat
```

```
   (48,58)        1
   (83,59)        1
   (57,60)        1
   (40,61)        1
   (61,62)        1
   (67,63)        1
   (67,64)        1
   (67,65)        1
   (67,66)        1
   (67,67)        1
   (67,68)        1
   (67,69)        1
   (67,70)        1
   (67,71)        1
   (67,72)        1
   (67,73)        1
   (67,74)        1
   (67,75)        1
    (3,76)        1
   (82,77)        1
   (45,78)        1
   (50,78)        1
   (45,79)        1
   (50,79)        1
```

So if you look at this model.rxnGeneMat that is one place where this information is, I do not think it is there in rules. Yeah, rules also have this information. So you have several reactions that have no rules basically empty rules and this means that you know gene 31 and gene 32 or gene 20 and gene 21 are together necessary right, either these are necessary together or these are necessary together for the reaction to happen.

So the hasEffect is very useful, so when you delete 32, it will actually check if it has an effect right. At the gene level, it may have an effect when you remove a gene but may be it affects no reaction right. So if you remove x5, it is not going to have any effect assuming that it is not there individually in some other reaction elsewhere right.

(Refer Slide Time: 03:54)

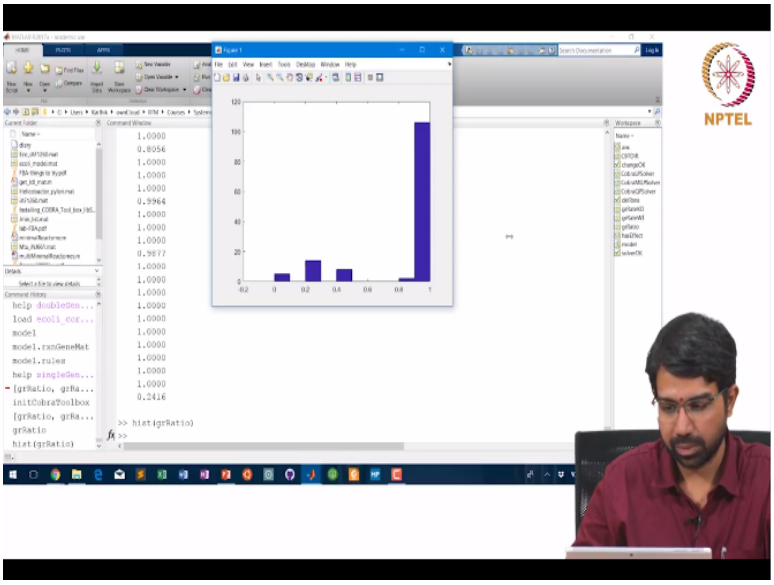


```
    (14,137)        1
    (58,137)        1
    (69,137)        1
    (70,137)        1

>> model.rules

ans =

  95×1 cell array

    '(x(6) | x(35))'
    'x(137)'
    '(x(97) | x(76) | x(57))'
    '(x(5) | x(36))'
    '(x(5) | x(36))'
    ''
    'x(9)'
    '( x(4)  &  x(16)  &  x(17) )'
    'x(87)'
    '(x(7) | x(39) | x(35))'
    ''
    '(((x(112) & x(113) & x(114)) & (x(107) & x(108) & x(109) & x(110) & x(111))) | ((x(112) & x(113)
    ''
```

If no x5, there will be no effect because x36 will be able to take care of that reaction. Now hasEffect is one of the outputs, you look at help single gene deletion, the last output is hasEffect. **"Professor - student conversation starts."** No, if the reactions are going to change, so look at the help so it says does the gene deletion affect anything that is our reactions removed from the model or not.

And you can use either FBA or MoMA and so on. It is, so it gives you the growth rate as well and the general assumption is something is lethal if it is<5% of the wild type growth rate. So if you look at rate such as 99.64% of wild type or 0.8 of wild type, 0.42 and yeah for some there is no 0, for many you find it is 1.0, you find a few 0s as well.

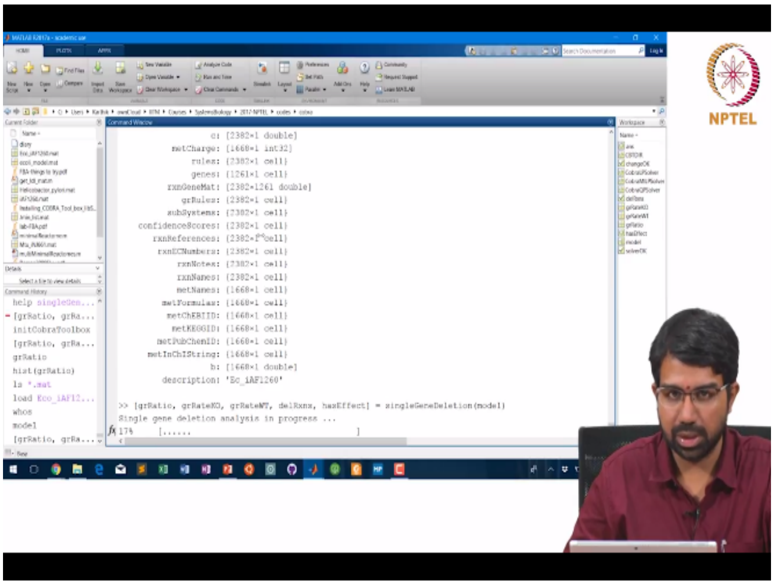
(Refer Slide Time: 05:23)



```
    1.0000
    0.8056
    1.0000
    1.0000
    1.0000
    0.9964
    1.0000
    1.0000
    1.0000
    0.9877
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    0.2416

>> hist(grRatio)
>>
```

So this is the distribution, there are few lethals, few sick deletions, few very lightly sick deletions and many non-lethal deletions. So none of those are lethal so this is the E. coli model, so let us just do the same thing on the E. coli model. **"Professor - student conversation ends."**

(Refer Slide Time: 06:03)

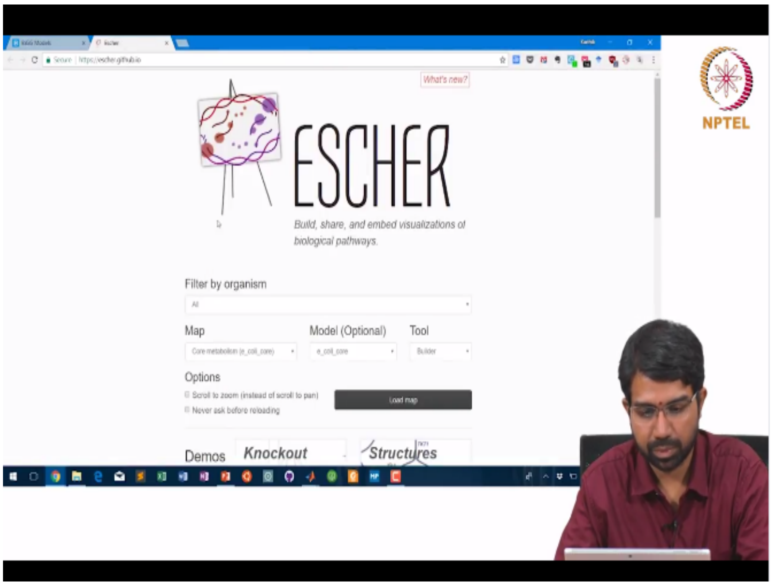


```
help singleGen...
[grRatio, grRa...
initCobraToolbox
[grRatio, grRa...
grRatio
hist(grRatio)
ls *.mat
load Eco_iAF12...
whos
model
[grRatio, grRa...
```

```
                  c: [2382×1 double]
          metCharge: [1668×1 int32]
              rules: [2382×1 cell]
              genes: [1261×1 cell]
         rxnGeneMat: [2382×1261 double]
            grRules: [2382×1 cell]
         subSystems: [2382×1 cell]
   confidenceScores: [2382×1 cell]
      rxnReferences: [2382×1 cell]
       rxnECNumbers: [2382×1 cell]
           rxnNotes: [2382×1 cell]
           rxnNames: [2382×1 cell]
           metNames: [1668×1 cell]
        metFormulas: [1668×1 cell]
         metChEBIID: [1668×1 cell]
          metKEGGID: [1668×1 cell]
       metPubChemID: [1668×1 cell]
     metInChIString: [1668×1 cell]
                  b: [1668×1 double]
        description: 'Ec_iAF1260'

>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleGeneDeletion(model)
Single gene deletion analysis in progress ...
17%      [......                              ]
```

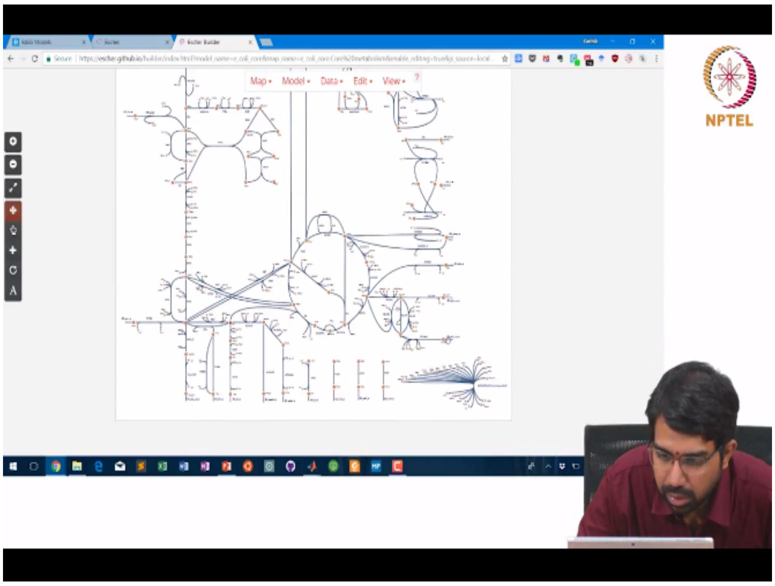
It is going to take a little longer to the single gene deletion because there are about 1260 genes in the E. coli model. We can download this model from the BIGG database. You will see that double gene deletion is like much, much lower the group force so you will have to use something like FastSL. While this is going on let me show you this nice web page.

(Refer Slide Time: 06:38)



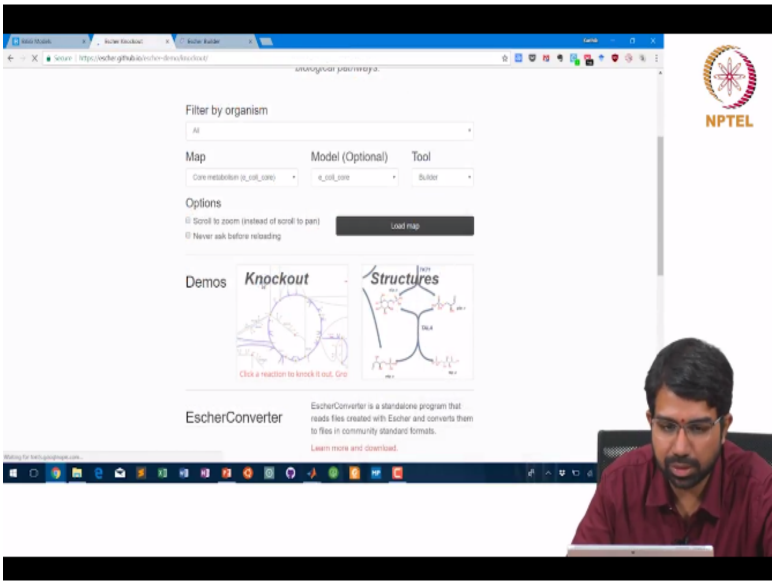
Right here it is called Escher.

(Refer Slide Time: 06:50)



This is a visualization of E. coli and you can see what happens when you start deleting reactions. You delete a reaction here, there is a way to knock out the reactions let me check.

(Refer Slide Time: 07:41)


Filter by organism: All

Map: Core metabolism (e_coli_core)

Model (Optional): e_coli_core

Tool: Builder

Options

☐ Scroll to zoom (instead of scroll to pan)

☐ Never ask before reloading

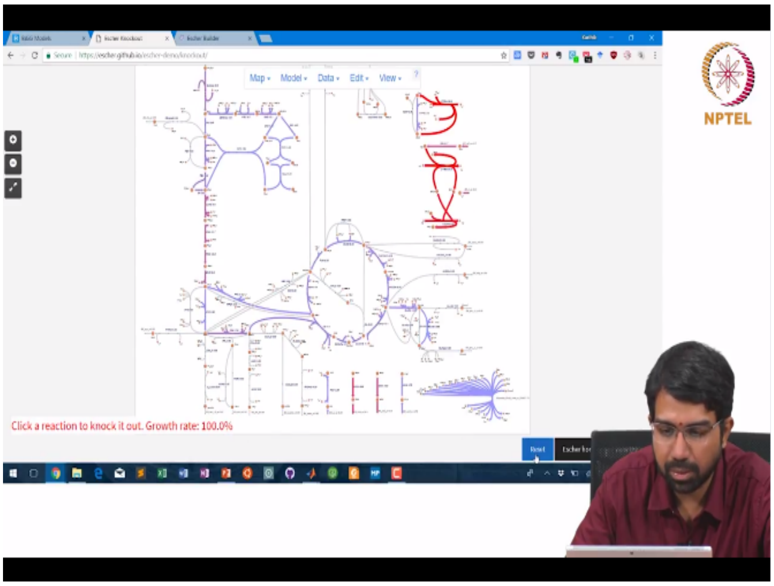
Click a reaction to knock it out.

EscherConverter is a standalone program that reads files created with Escher and converts them to files in community standard formats.

Learn more and download.

Then you have knock outs of reactions on Escher online, so you just a click a reaction to knock it out.

(Refer Slide Time: 07:50)



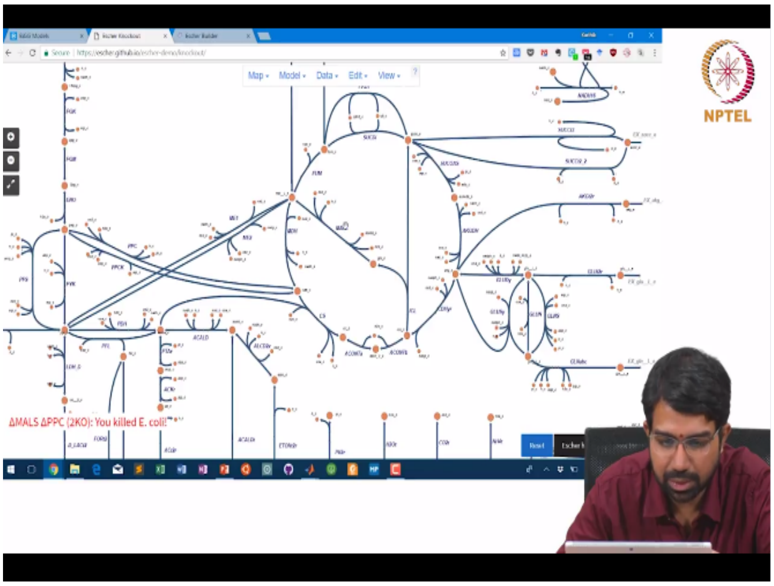
Click a reaction to knock it out. Growth rate: 100.0%

So inaugural from succinate dehydrogenase reaction or something. I have knocked out one more thing and growth rate is now 73%. Let us try to knock out something else, 3 knockouts it has gone to 28%, 4 knockouts it has gone to 26.5%, 5th knockout it killed E. coli. So this is like a quintuple synthetic lethal set right. So if you remove them separately, nothing happens. When you remove all 5 will not necessarily it could also have been that this is singly lethal.

Yeah, but you know for sure that these are all not single lethals right. **"Professor - student conversation starts."** They must have done the similarity check there, I do not know what is the exact cut off they used. So this growth rate is 99.1, yeah so this might be a single lethal. So if you see I can keep deleting these reactions which are not going to make an effect. These are not reactions that are currently been used right but however if I even delete one of these it might already become lethal. **"Professor - student conversation ends."**

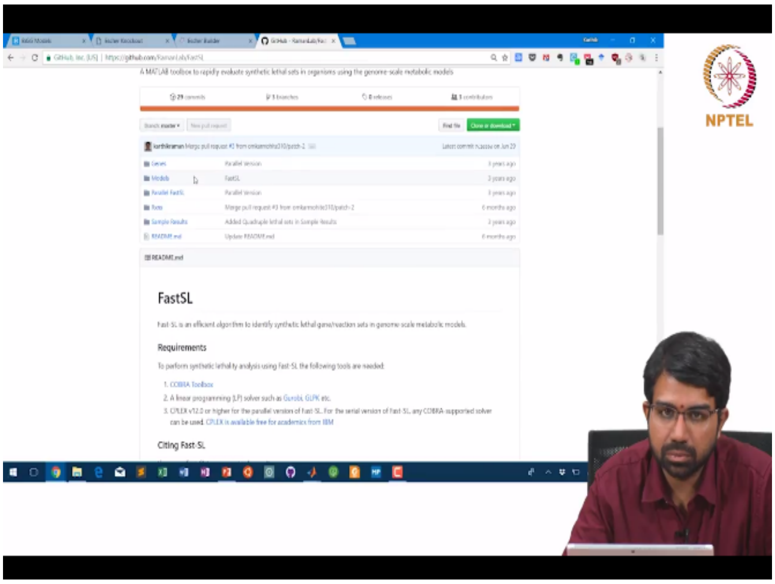
Yeah, so this was a nice example. So I kept deleting multiple genes nothing happened but then finally when I deleted one I think we have a synthetic lethal here, delta PPC and MALS looks to be a synthetic lethal.

(Refer Slide Time: 11:17)



Yeah, this is a synthetic lethal, so PPC and MALS.

(Refer Slide Time: 11:55)



A MATLAB toolbox to rapidly evaluate synthetic lethal sets in organisms using the genome-scale metabolic models

Genes	Parallel Version	3 years ago
Models	FastSL	3 years ago
Parallel FastSL	Parallel Version	3 years ago
Tests	Merge pull request #3 from omkarmohite310/patch-2	6 months ago
Sample Results	Added Quadruple lethal sets in Sample Results	3 years ago
README.md	Update README.md	6 months ago

FastSL

Fast-SL is an efficient algorithm to identify synthetic lethal gene/reaction sets in genome scale metabolic models.

Requirements

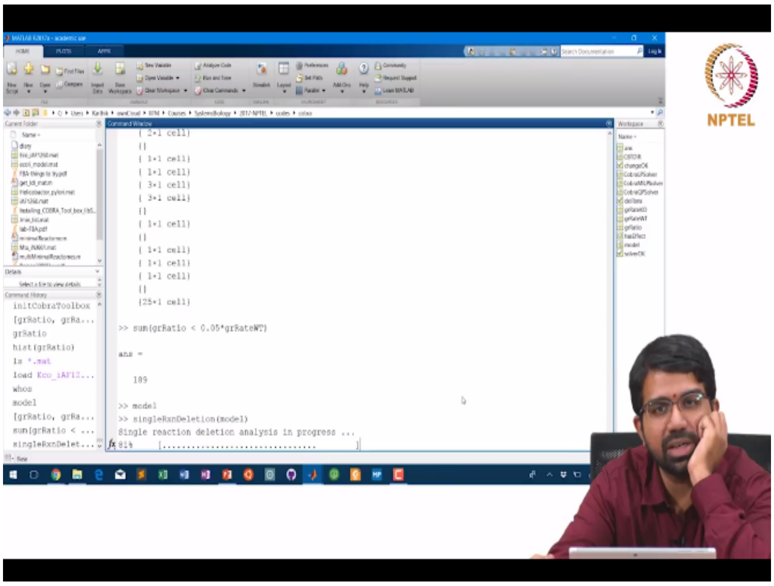
To perform synthetic lethality analysis using Fast-SL the following tools are needed:

1. COBRA Toolbox
2. A linear programming (LP) solver such as Gurobi, GLPK etc.
3. CPLEX v12.0 or higher for the parallel version of Fast-SL. For the serial version of Fast-SL, any COBRA-supported solver can be used. CPLEX is available free for academics from IBM

Citing Fast-SL

So if you download FastSL, you will also get these example matrices I mean MATLAB files which includes the stoichiometric matrix and the entire model actually. **“Professor - student conversation starts.”** That is not integrated with COBRA but the code is here, the code is (()) (12:28). So this is how you download a separate package and install it. **“Professor - student conversation ends.”** So how many single lethals in E. coli, how do you do it?

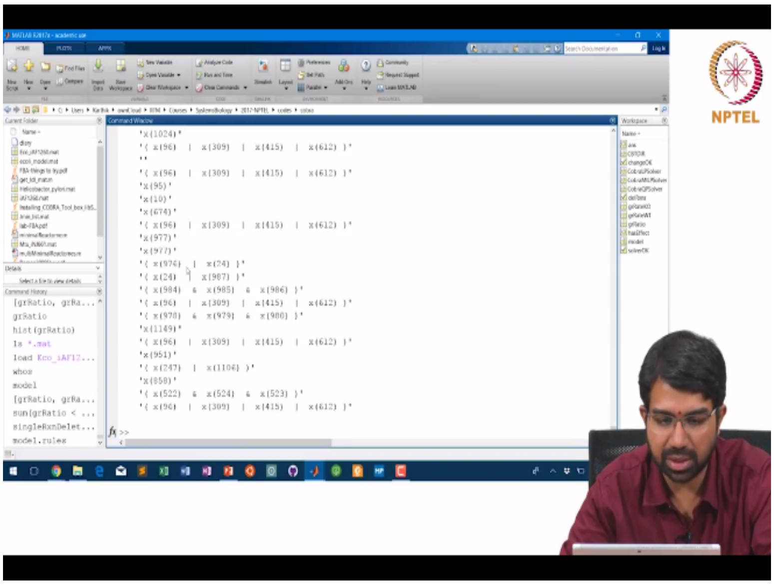
(Refer Slide Time: 13:05)



```
  { 2×1 cell}
  []
  { 1×1 cell}
  { 1×1 cell}
  { 3×1 cell}
  { 3×1 cell}
  []
  { 1×1 cell}
  []
  { 1×1 cell}
  { 1×1 cell}
  { 1×1 cell}
  []
  {25×1 cell}

>> sum(grRatio < 0.05*grRateWT)

ans =

   189

>> model
>> singleRxnDeletion(model)
Single reaction deletion analysis in progress ...
81%    [..............................         ]
```

Now can you try out single reaction deletion? Is it much faster than what it is for me? **"Professor - student conversation starts."** So reaction will remove reaction by reaction, gene will remove gene by gene and delete the corresponding reactions. Some genes if removed may not lead to the removal of any reaction. If gene 1 is not connected to any other reaction, if reaction 1 has a GPR which says like these are here right I think it is almost over.

(Refer Slide Time: 14:12)



```
    'x(1024)'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    ''
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    'x(95)'
    'x(10)'
    'x(674)'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    'x(977)'
    'x(977)'
    '( x(976)  |  x(24) )'
    '( x(24)  |  x(987) )'
    '( x(984)  &  x(985)  &  x(986) )'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    '( x(978)  &  x(979)  &  x(980) )'
    'x(1149)'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    'x(951)'
    '( x(247)  |  x(1106) )'
    'x(858)'
    '( x(522)  &  x(524)  &  x(523) )'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'

>>
```

So model.rules yeah so this is 976 or 24, so if you remove 24 nothing will happen most likely. It would not have it okay, so this is an interesting point. So how does FBA work? It tries to maximize the growth rate through every reaction right. So it does not actually try to account if there are like multiple copies of a gene you can have a higher growth rate or any of those things right.

So if that happens see if you overexpress a gene, you can like basically you know double that value or something like that. So all this is regulatory information right which is not there in plane FBA. In future lecture, we will look at E-Flux which will essentially use this method. E-flux you know appropriately waits different changes the lb and ub for reactions depending upon the expression levels of the genes.

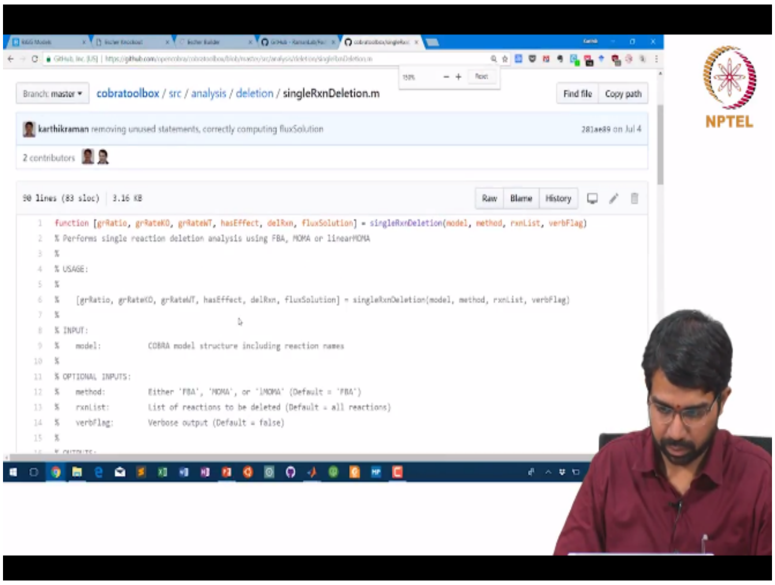
It helps you integrate say microarray data or any expression data set that you have with your flux model to make better predictions. So there are many methods to integrate, E-Flux is one of them. **"Professor - student conversation ends."** I am just running this again and let us see what hasEffect looks like? So do you have an answer for the last question? What would be the minimal set of non-essential reactions in a given model?

"Professor - student conversation starts." More or less yes you are in the right track. (()) (16:30) telling today in morning's class. Good point, yes and no. The answer is no, why? Whether a reaction is known or not, so you might be removing a non-essential reaction. No, right so that again see your previous approach is little bit, Lavanya you have an answer, you did this already right.

That is a slow in blocked reactions but beyond that you can still remove some more reactions. That is just like that is read on it. It will just give you r1, r2, r3, r4, rn. Yeah, for reaction there is no question of doing hasEffect, so does reaction have it I was trying to remove it out of COBRA toolbox. It does have, what is the documentation say? It is just going to be all 1s. Look at the value of hasEffect, I think it is just going to be all 1s.

I did not try to edit it out of the toolbox, it is like we argued and was redundant but we may be kept it in for like backward compatibility. **"Professor - student conversation ends."**

(Refer Slide Time: 18:45)



cobratoolbox / src / analysis / deletion / singleRxnDeletion.m

karthikraman removing unused statements, correctly computing fluxSolution 281ae89 on Jul 4

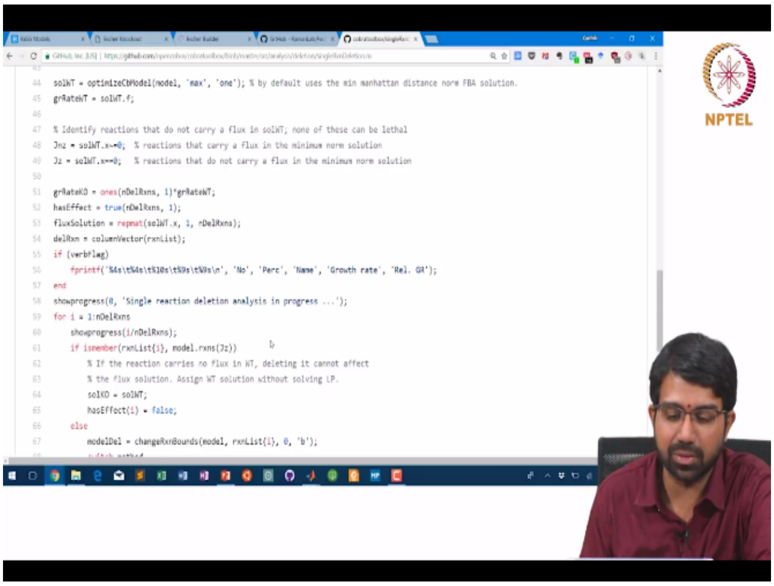
2 contributors

90 lines (83 sloc) | 3.16 KB

```
function [grRatio, grRateKO, grRateWT, hasEffect, delRxn, fluxSolution] = singleRxnDeletion(model, method, rxnList, verbFlag)
% Performs single reaction deletion analysis using FBA, MOMA or linearMOMA
%
% USAGE:
%
%    [grRatio, grRateKO, grRateWT, hasEffect, delRxn, fluxSolution] = singleRxnDeletion(model, method, rxnList, verbFlag)
%
% INPUT:
%    model:           COBRA model structure including reaction names
%
% OPTIONAL INPUTS:
%    method:          Either 'FBA', 'MOMA', or 'lMOMA' (Default = 'FBA')
%    rxnList:         List of reactions to be deleted (Default = all reactions)
%    verbFlag:        Verbose output (Default = false)
%
```

So this is the code for single reaction deletion right and this is based on FastSL that we discussed a little earlier.

(Refer Slide Time: 19:02)



```
solWT = optimizeCbModel(model, 'max', 'one'); % by default uses the min manhattan distance norm FBA solution.
grRateWT = solWT.f;

% Identify reactions that do not carry a flux in solWT; none of these can be lethal
Jnz = solWT.x~=0;  % reactions that carry a flux in the minimum norm solution
Jz = solWT.x==0;   % reactions that do not carry a flux in the minimum norm solution

grRateKO = ones(nDelRxns, 1)*grRateWT;
hasEffect = true(nDelRxns, 1);
fluxSolution = repmat(solWT.x, 1, nDelRxns);
delRxn = columnVector(rxnList);
if (verbFlag)
    fprintf('%4s\t%4s\t%10s\t%9s\t%9s\n', 'No', 'Perc', 'Name', 'Growth rate', 'Rel. GR');
end
showprogress(0, 'Single reaction deletion analysis in progress ...');
for i = 1:nDelRxns
    showprogress(i/nDelRxns);
    if ismember(rxnList{i}, model.rxns(Jz))
        % If the reaction carries no flux in WT, deleting it cannot affect
        % the flux solution. Assign WT solution without solving LP.
        solKO = solWT;
        hasEffect(i) = false;
    else
        modelDel = changeRxnBounds(model, rxnList{i}, 0, 'b');
```

So here the key ideas here so if the reaction carries no flux in wild type, deleting it cannot affect the flux solution. See not all reactions will carry a flux in the wild type solution right. So corresponding to Biomass optimization, for any objective function you run FBA and you get a flux solution right a flux distribution. **"Professor - student conversation starts."** Exactly, right so there are many equivalent flux distributions one this that is guaranteed for you is that the function value is the same right.

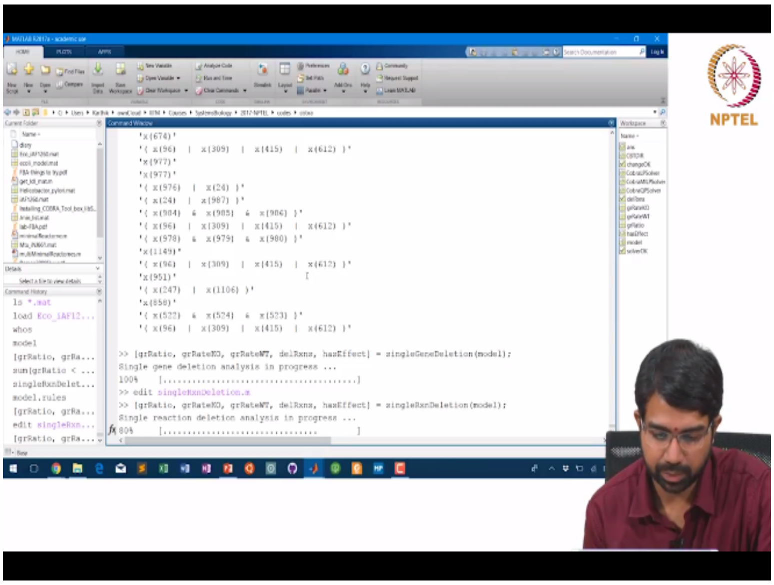
So out of these equivalent flux distributions which of them you know so can you use that to find out which reactions are essential and not essential? So there are some reactions which always stay 0 right like your blocked reaction, so you can be done a flux variability analysis and if both the lower and upper bounds are I mean the maximum and minimum values are 0, you can discard those reactions but beyond that even out of a pair of reactions like we were seeing here, let me just reset this.

You see that only MALS is not currently carrying a flux right but it could very well be carrying a flux. If I delete this right the growth rate has not changed for practical purposes right and this is carrying a flux instead right. So now if I go and touch this it gets you know the knocked out and you have a synthetic lethal right but basically which of these do you pick initially? It is almost arbitrary it depends upon the solver, both these are equivalent states in which the cell can exist in reality right.

So to come back let us look at this part of the code now. So if the reaction carries no flux in the wild type right in a FBA solution in any given FBA solution you can remove it without affecting the flux solution and you can basically assign the wild type solution to the knock out solution without even computing or solving the LP and that is where we say that hasEffect is false.

There is no effect actually on deleting that reaction. So if you run this, you will now see this is much faster. **"Professor - student conversation ends."**

(Refer Slide Time: 22:04)

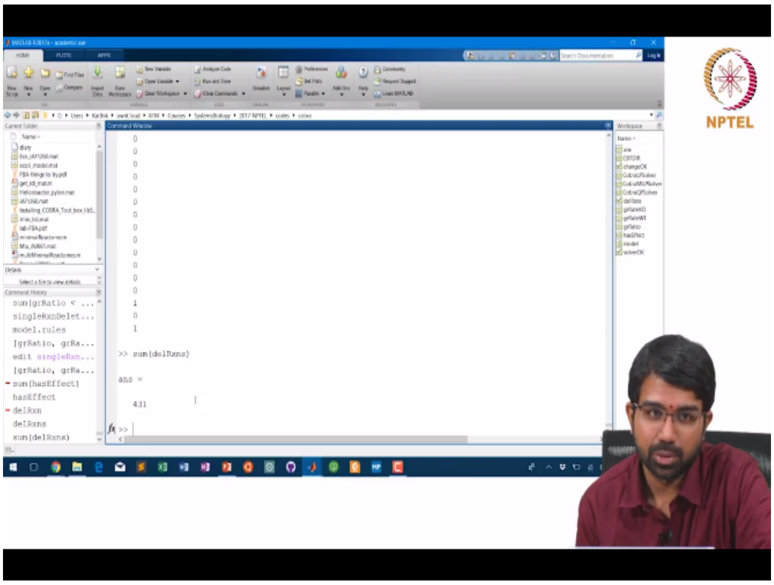


```
    'x(674)'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    'x(977)'
    'x(977)'
    '( x(976)  |  x(24) )'
    '( x(24)  |  x(987) )'
    '( x(984)  &  x(985)  &  x(986) )'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    '( x(978)  &  x(979)  &  x(980) )'
    'x(1149)'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'
    'x(951)'
    '( x(247)  |  x(1106) )'
    'x(858)'
    '( x(522)  &  x(524)  &  x(523) )'
    '( x(96)  |  x(309)  |  x(415)  |  x(612) )'

>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleGeneDeletion(model);
Single gene deletion analysis in progress ...
100%    [.........................................]
>> edit singleRxnDeletion.m
>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleRxnDeletion(model);
Single reaction deletion analysis in progress ...
80%     [..................................        ]
```

```
ls *.mat
load Eco_iAF12...
whos
model
[grRatio, grRa...
sum(grRatio < ...
singleRxnDelet...
model.rules
[grRatio, grRa...
edit singleRxn...
[grRatio, grRa...
```

Yeah, you can see this is way faster than the deletion that I was running previously, single reaction deletion right. So now oops I think it has a reaction delRxn are swapped.

(Refer Slide Time: 23:10)

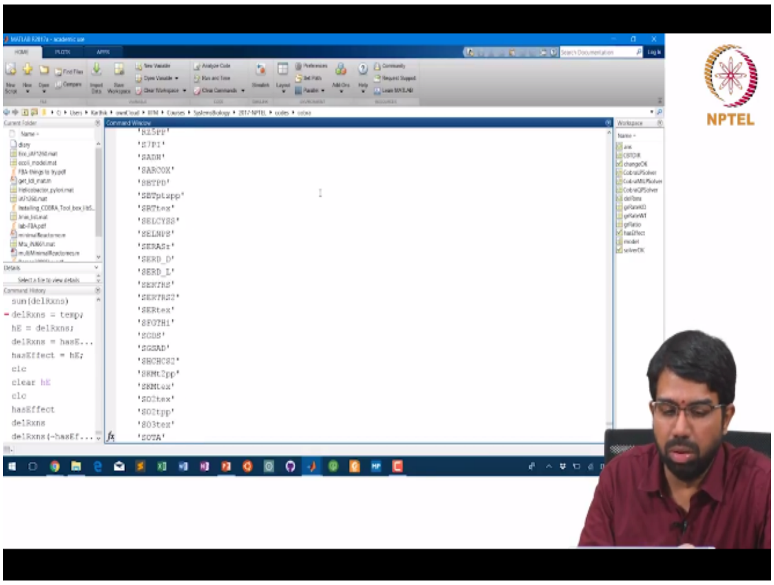


```
0
0
0
0
0
0
0
0
0
0
0
0
0
1
0
1

>> sum(delRxns)

ans =

   431

>>
```

```
sum(grRatio < ...
singleRxnDelet...
model.rules
[grRatio, grRa...
edit singleRxn...
[grRatio, grRa...
sum(hasEffect)
hasEffect
delRxn
delRxns
sum(delRxns)
```

I thought I fix this anyway, so there are 431 reactions that cannot be knocked out so if I start with a subset of this reaction set so let us just **"Professor - student conversation starts."** No, no, no this was like variable mismatch. This is actually hasEffect. The code has an error, oh god I have to put a pull request for this. No, not all that is exactly precisely what I was trying to tell you all along right.

(Refer Slide Time: 24:45)



```
'RZ5PP'
'S7PI'
'SADH'
'SARCOX'
'SBTPD'
'SBTptspp'
'SBTtex'
'SELCYSS'
'SELNPS'
'SERASr'
'SERD_D'
'SERD_L'
'SERTRS'
'SERTRS2'
'SERtex'
'SFGTHi'
'SGDS'
'SGSAD'
'SHCHCS2'
'SHKt2pp'
'SHKtex'
'SO2tex'
'SO2tpp'
'SO3tex'
'SOTA'
```

```
sum(delRxns)
delRxns = temp;
hE = delRxns;
delRxns = hasE...
hasEffect = hE;
clc
clear hE
clc
hasEffect
delRxns
delRxns(~hasEf...
```

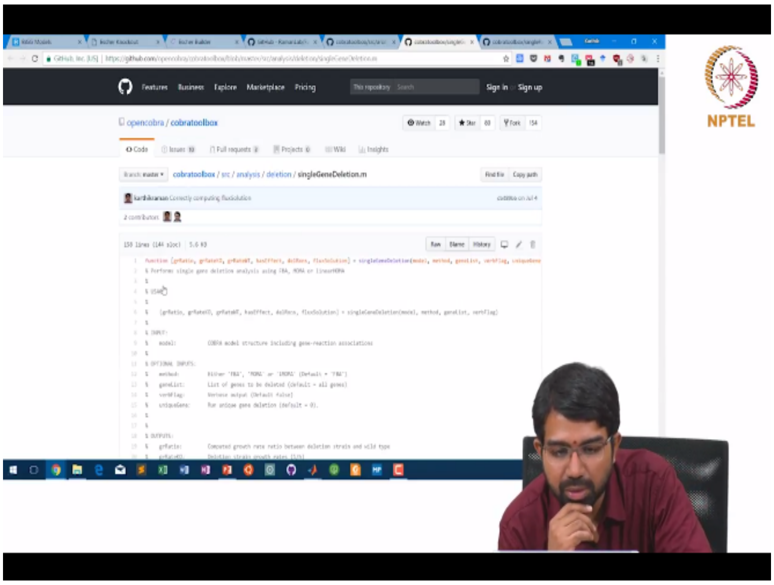
So some reactions okay so let us just see okay wait. So all these reactions do not have an effect, when you delete them nothing happens. So that reaction is not carrying a flux, it is not currently important for the cell. So there are two, three possibilities. When you delete a reaction and nothing happens, there are three possibilities. What are the three possibilities? We discussed them before at least two of them we clearly discussed them before.

Yeah but that is like just saying you know A=B, B=A right. So I want you to tell what is the reason that that happens, it could be blocked reactions, it is not like a major name. It could be a conditionally blocked reaction or it could be a gap or an unconditionally blocked reaction. So these are two possibilities. The third possibility is that it is not lethal but that is also a possibility, it could be not lethal I do not know if that is some usually exclusive cases not lethal.

The fourth possibility is that it could be part of a higher lethal set. When you remove a reaction out of a triple lethal set, nothing is going to happen. If you remove two reactions out of a triple lethal set nothing is going to happen. Only when you remove all three, the organism is going to die. Who is referring to? HasEffect basically says removing your reaction changes has an effect on the growth rate or not.

No, not lethally well whether it even changes growth rate or whether it even changes the reaction flux, whether it is even important, whether it is unblocked so it is actually a slightly debatable thing. HasEffect for gene is the much easier thing to understand. The code is not wrong. I had given the arguments in the wrong order here. **"Professor - student conversation ends."** The gene deletion and reaction deletion orders are different, let us just see.

(Refer Slide Time: 27:50)

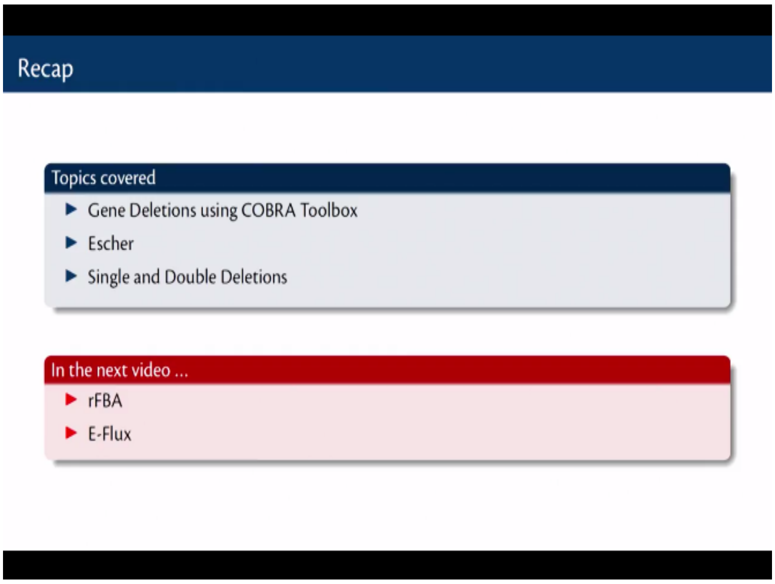


```
function [grRatio, grRateKO, grRateWT, hasEffect, delRxns, fluxSolution] = singleGeneDeletion(model, method, geneList, verbFlag, uniqueGene
% Performs single gene deletion analysis using FBA, MOMA or linearMOMA
%
% USAGE:
%
%    [grRatio, grRateKO, grRateWT, hasEffect, delRxns, fluxSolution] = singleGeneDeletion(model, method, geneList, verbFlag)
%
% INPUT:
%    model:          COBRA model structure including gene-reaction associations
%
% OPTIONAL INPUTS:
%    method:         Either 'FBA', 'MOMA' or 'lMOMA' (Default = 'FBA')
%    geneList:       List of genes to be deleted (default = all genes)
%    verbFlag:       Verbose output (Default false)
%    uniqueGene:     Run unique gene deletion (default = 0).
%
%
% OUTPUTS:
%    grRatio:        Computed growth rate ratio between deletion strain and wild type
%    grRateKO:       Deletion strain growth rates (1/h)
```

This is growth ratio, growth rate KO, growth rate wild type, hasEffect, del reactions. This is growth ratio, growth rate KO, growth rate wild type, hasEffect, del reactions. It is similar maybe okay maybe the version I had was slightly different. This is del reactions hasEffect that is where the problem is okay. So where are we, how much of this is done? Let me try all of this, this is easy, seriously easy stuff.

You have to learn out your delete model genes. What will be the naïve way to do this? Find the minimal non-essential reactions. The naïve way to do is to remove a reaction. If it does not kill the organism, keep iterate. Then, remove another reaction, if it kills the organism, revert that deletion repeat.

(Refer Slide Time: 29:36)



Recap

Topics covered

- Gene Deletions using COBRA Toolbox
- Escher
- Single and Double Deletions

In the next video ...

- rFBA
- E-Flux

So in today's lab, we had an overview of how we perform gene deletions using the COBRA toolbox. I also showed you this interesting tool called Escher which gives you a visual picture of what happens when you delete genes and we can also see when the organism stops growing and so on and we also looked at single and double gene deletions.

In the next video, we will see how to integrate regulatory information into flux balance models and get better predictions. We will look at two techniques, one called regulated flux balance analysis or rFBA and the other called E-Flux which gives a nice way to constraint the reactions based on transcription data or transcriptomic data.