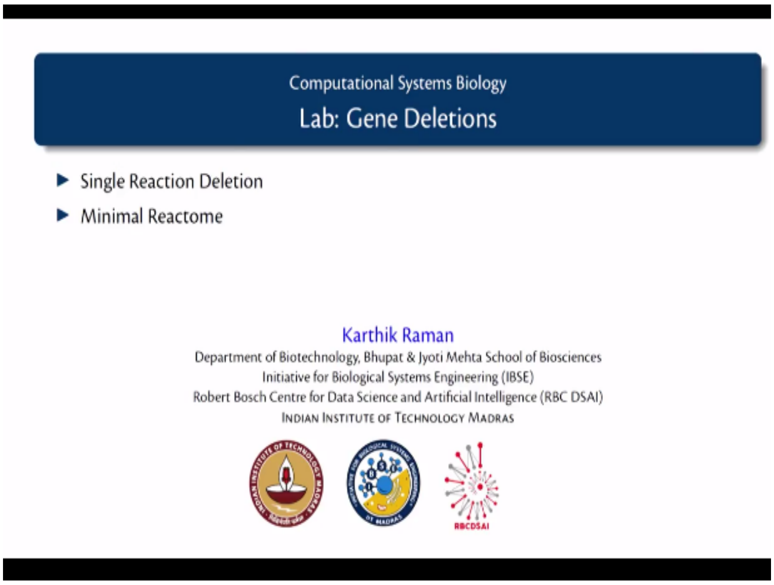
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology – Madras

Lecture - 78
Lab: Gene Deletions

So welcome to today’s lab session.

(Refer Slide Time: 00:13)



Computational Systems Biology

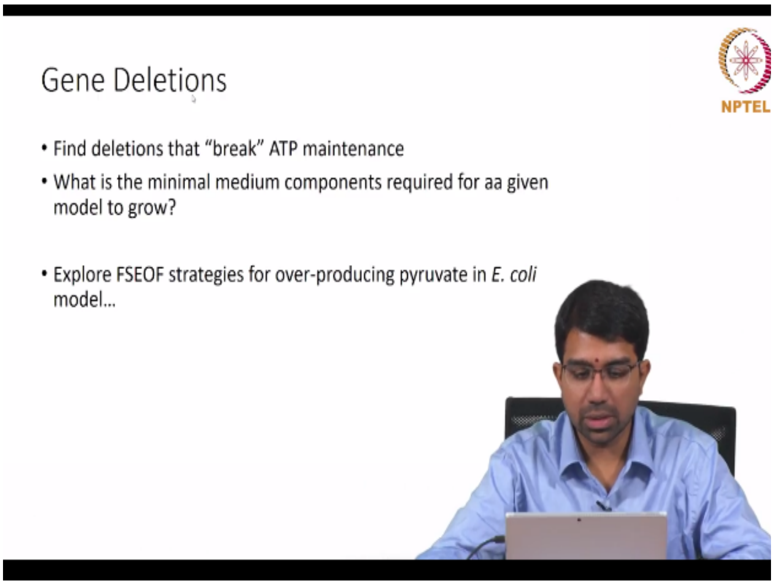
Lab: Gene Deletions

- Single Reaction Deletion
- Minimal Reactome

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

We will look at gene deletions, we will look at single reaction or single gene deletion and we will also look at how do we build up the minimal reactome for a given organism. So what is the least set of reactions or what is the smallest set of reactions that need to carry a flux for an organism to survive and this has interesting off-shoots from the synthetic lethality analysis that we were talking about a few videos earlier.

(Refer Slide Time: 00:38)



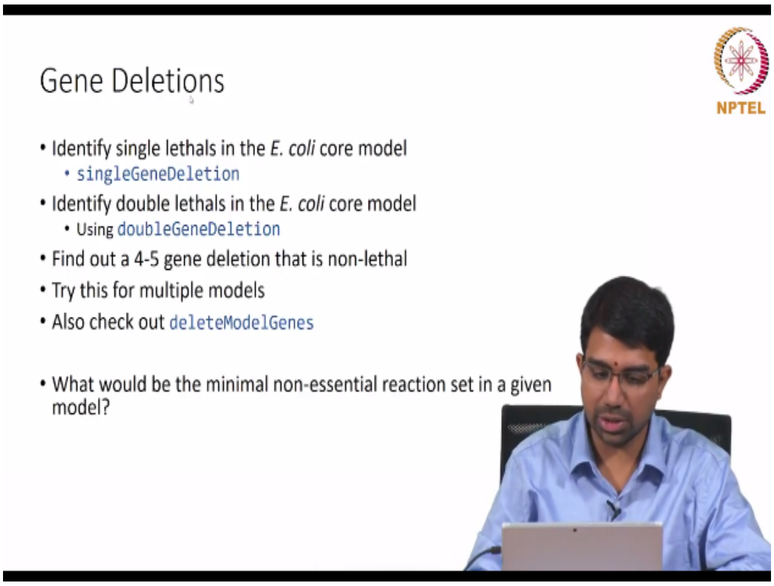
Gene Deletions

- Find deletions that "break" ATP maintenance
- What is the minimal medium components required for aa given model to grow?
- Explore FSEOF strategies for over-producing pyruvate in *E. coli* model…

So let us continue with some lab exercises in constraint based modelling. So can we now identify gene deletions that break the ATP maintenance flux. So do we get infeasible solution and so on, let us explore that and for any given model how would you identify the minimum set of medium components for it to grow. Look for example what are the essential amino acids or something like that.

And can you explore FSEOF strategies for over producing say pyruvate in E coli model, it maybe growth associated likely but let us just figure out if we can come with FSEOF strategies. So let us try to solve these.

(Refer Slide Time: 01:14)

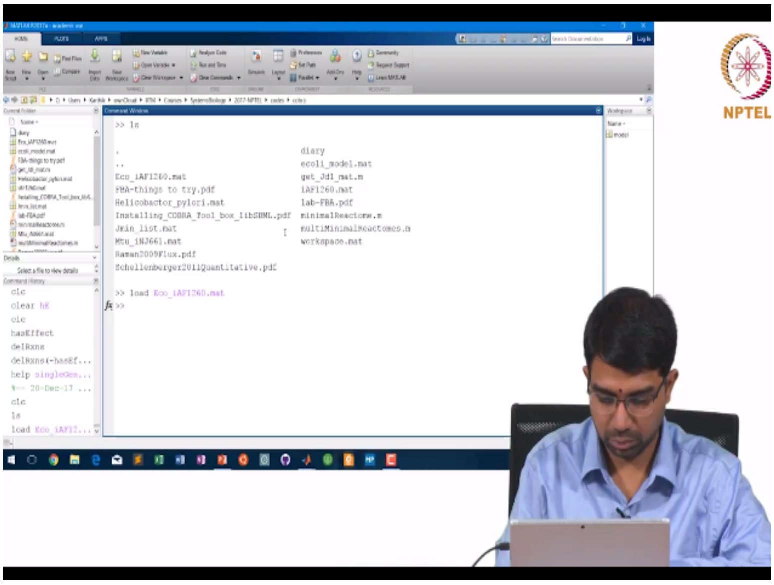


Gene Deletions

- 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?

In addition, we can also revisit some of these which we were looking at in the previous lab session.

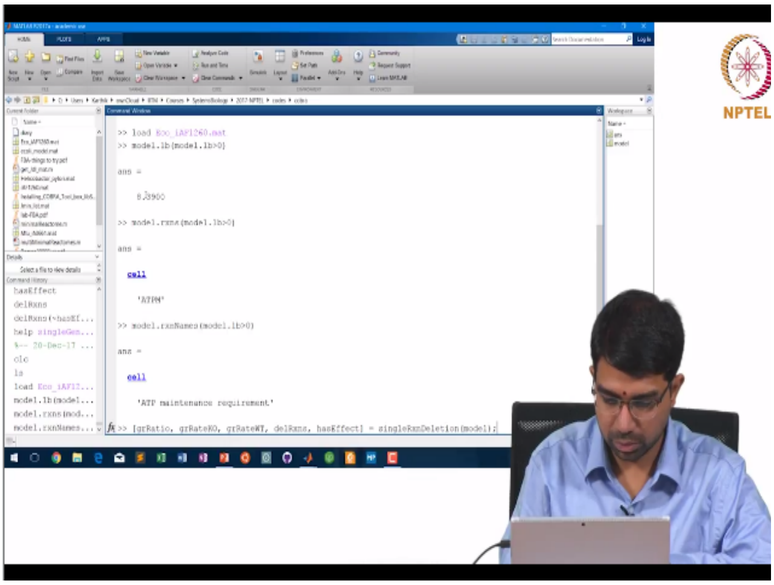
(Refer Slide Time: 02:00)



```
>> ls

.                                         diary
..                                        ecoli_model.mat
Eco_iAF1260.mat                           get_Jdl_mat.m
FBA-things to try.pdf                     iAF1260.mat
Helicobactor_pylori.mat                   lab-FBA.pdf
Installing_COBRA_Tool_box_libSBML.pdf     minimalReactome.m
Jmin_list.mat                             multiMinimalReactomes.m
Mtu_iNJ661.mat                            workspace.mat
Raman2009Flux.pdf
Schellenberger2011Quantitative.pdf

>> load Eco_iAF1260.mat
>>
```

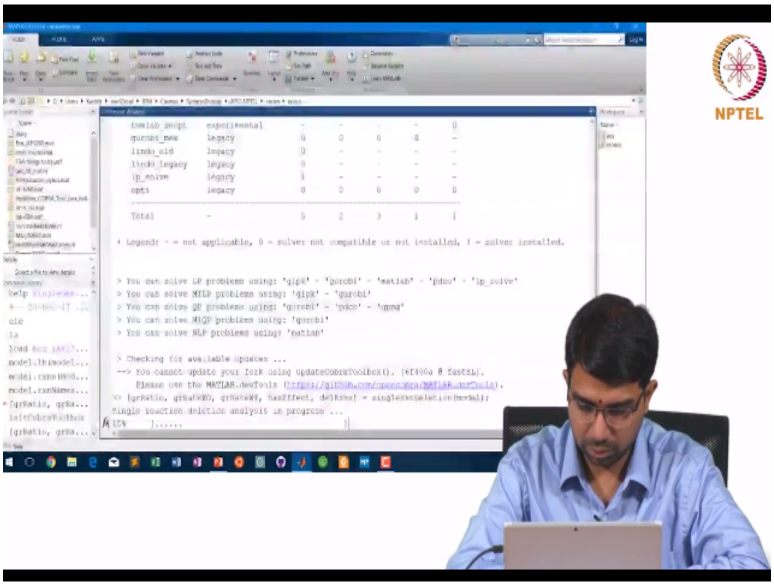
So if you see there is only 1 lb which is > 0, everything else is 0 or less and that corresponds to ATP maintenance just 8.39. **"Professor - student conversation starts"** So if we just keep on deleting reactions and see whether they will fit (()) (02:46) becomes 0 or not. There is probably one way to look at it because I do not think single gene deletion will directly show you that.

(Refer Slide Time: 03:00)



```
>> load Eco_iAF1260.mat
>> model.lb(model.lb>0)

ans =

    8.3900

>> model.rxns(model.lb>0)

ans =

  cell

    'ATPM'

>> model.rxnNames(model.lb>0)

ans =

  cell

    'ATP maintenance requirement'

>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleRxnDeletion(model);
```

(Refer Slide Time: 04:03)

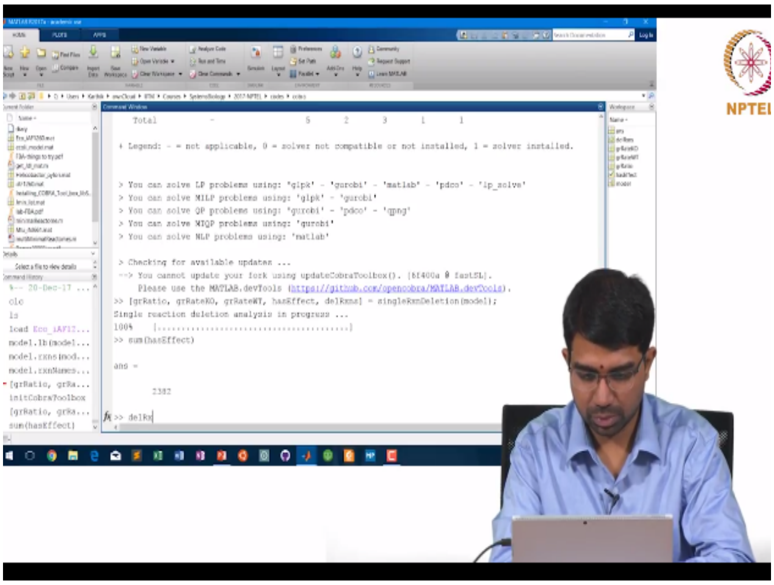


```
    tomlab_snopt    experimental      -      -      -      -      0
    gurobi_mex      legacy            0      0      0      0      -
    lindo_old       legacy            0      -      -      -      -
    lindo_legacy    legacy            0      -      -      -      -
    lp_solve        legacy            1      -      -      -      -
    opti            legacy            0      0      0      0      0
    ---------------------------------------------------------------------
    Total           -                 5      2      3      1      1

 + Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver installed.


 > You can solve LP problems using: 'glpk' - 'gurobi' - 'matlab' - 'pdco' - 'lp_solve'
 > You can solve MILP problems using: 'glpk' - 'gurobi'
 > You can solve QP problems using: 'gurobi' - 'pdco' - 'qpng'
 > You can solve MIQP problems using: 'gurobi'
 > You can solve NLP problems using: 'matlab'

 > Checking for available updates ...
 --> You cannot update your fork using updateCobraToolbox(). [6f400a @ fastSL].
     Please use the MATLAB.devTools (https://github.com/opencobra/MATLAB.devTools).
>> [grRatio, grRateKO, grRateWT, hasEffect, delRxns] = singleRxnDeletion(model);
Single reaction deletion analysis in progress ...
15%     [......
```

So how are you going about it. So have you already looked delete model genes in those commands. So try to find minimal genome right, by doing the delete model genes that is another exercise you can try out. So delete model genes that will delete (()) (04:58). That is how you will instead of putting the constraints by yourself you look through some genes and you delete them one by one or whatever.

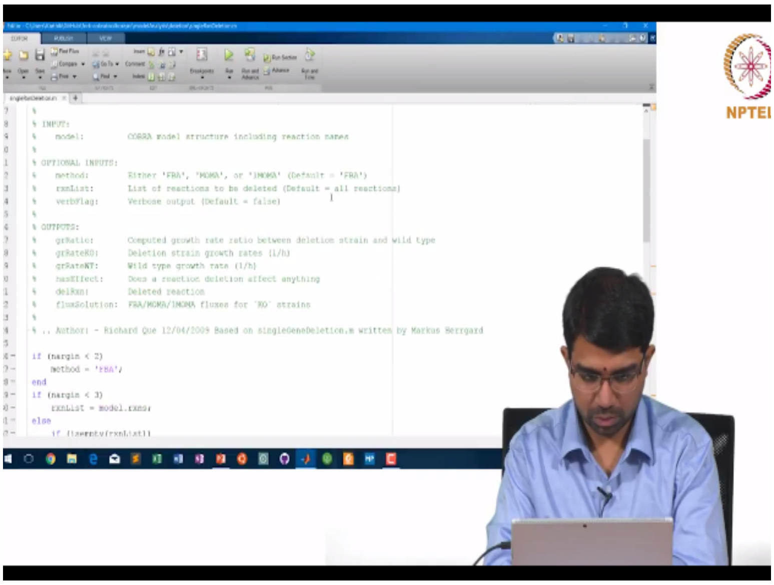
(()) (05:10) if we make a subset of all the things (()) (05:12) 2 power number of things right. You do not want to try all the subsets, you incrementally go deleting genes. If it dies you undo it and you redo that kind of thing. Yeah, so this has affect us like fully one, it is the older version of single reaction deletion.

(Refer Slide Time: 05:46)



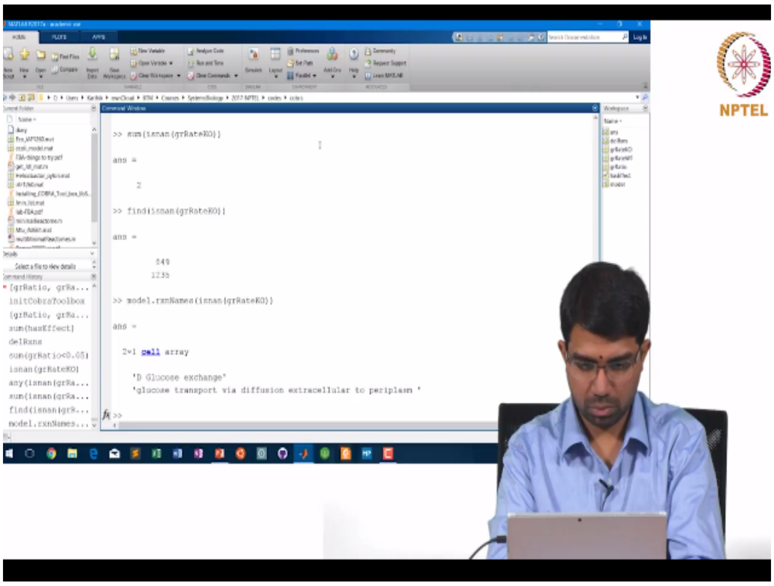
```
    Total           -                 5      2      3      1      1

 + Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver installed.


 > You can solve LP problems using: 'glpk' - 'gurobi' - 'matlab' - 'pdco' - 'lp_solve'
 > You can solve MILP problems using: 'glpk' - 'gurobi'
 > You can solve QP problems using: 'gurobi' - 'pdco' - 'qpng'
 > You can solve MIQP problems using: 'gurobi'
 > You can solve NLP problems using: 'matlab'

 > Checking for available updates ...
 --> You cannot update your fork using updateCobraToolbox(). [6f400a @ fastSL].
     Please use the MATLAB.devTools (https://github.com/opencobra/MATLAB.devTools).
>> [grRatio, grRateKO, grRateWT, hasEffect, delRxns] = singleRxnDeletion(model);
Single reaction deletion analysis in progress ...
100%    [........................................]
>> sum(hasEffect)

ans =

        2382

>> delRx
```

(Refer Slide Time: 06:45)



```
%
% 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)
%
% 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 reaction deletion affect anything
%    delRxn:        Deleted reaction
%    fluxSolution:  FBA/MOMA/lMOMA fluxes for `KO` strains
%
% .. Author: - Richard Que 12/04/2009 Based on singleGeneDeletion.m written by Markus Herrgard

if (nargin < 2)
    method = 'FBA';
end
if (nargin < 3)
    rxnList = model.rxns;
else
    if (isempty(rxnList))
```

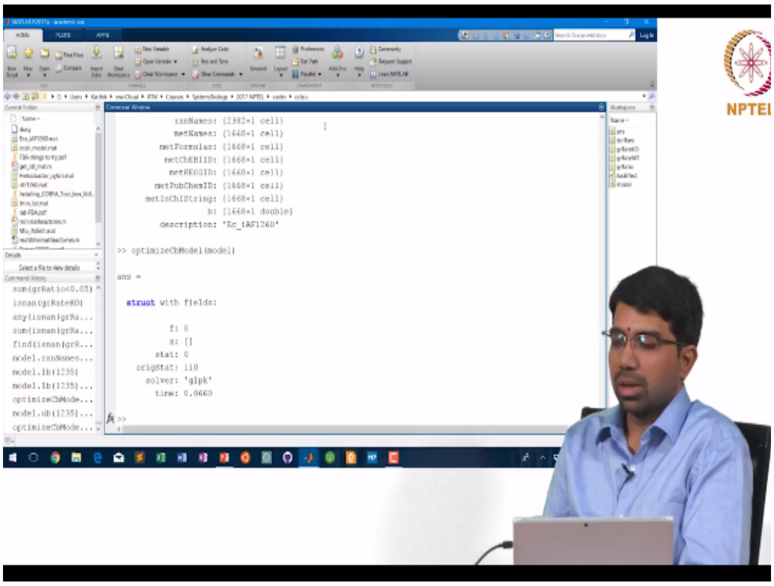
(Refer Slide Time: 07:45)



```
>> sum(isnan(grRateKO))

ans =

     2

>> find(isnan(grRateKO))

ans =

         549
        1235

>> model.rxnNames(isnan(grRateKO))

ans =

  2×1 cell array

    'D Glucose exchange'
    'glucose transport via diffusion extracellular to periplasm '

>>
```

In incremental deletion, so you are talking about incremental deletion for finding the minimal gene onset right, for ATP we cannot use deletion for finding the whether (()) (08:43).

(Refer Slide Time: 08:45)



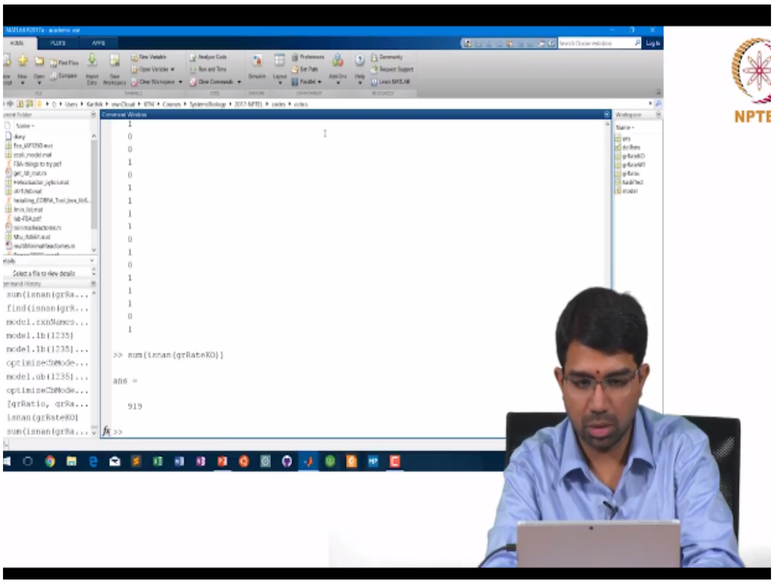
```
sum(grRatio<0.05)
isnan(grRateKO)
any(isnan(grRa...
sum(isnan(grRa...
find(isnan(grR...
model.rxnNames...
model.lb(1235)
model.lb(1235)...
optimizeCbMode...
model.ub(1235)...
optimizeCbMode...
```

```
          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'

>> optimizeCbModel(model)

ans =

  struct with fields:

          f: 0
          x: []
       stat: 0
   origStat: 110
     solver: 'glpk'
       time: 0.0660

fx >>
```

You can try random sets and see if it happens with single reaction deletion it does not seem to happen, I do not know about single gene deletion, maybe you should try because I distinctly remember that being an issue. Here (()) (08:59) number of gene subset will come. They are possible, but I am just asking you to explore, I not, see only if you have to evaluate all possible subsets right.

You can basically just simulate some subsets some fraction of genes. For every subject we have to check whether that the biomass is whether the cell is growing or not for every subset, yeah, so you are asking us to like find the first subset which, something like that.

(Refer Slide Time: 09:45)

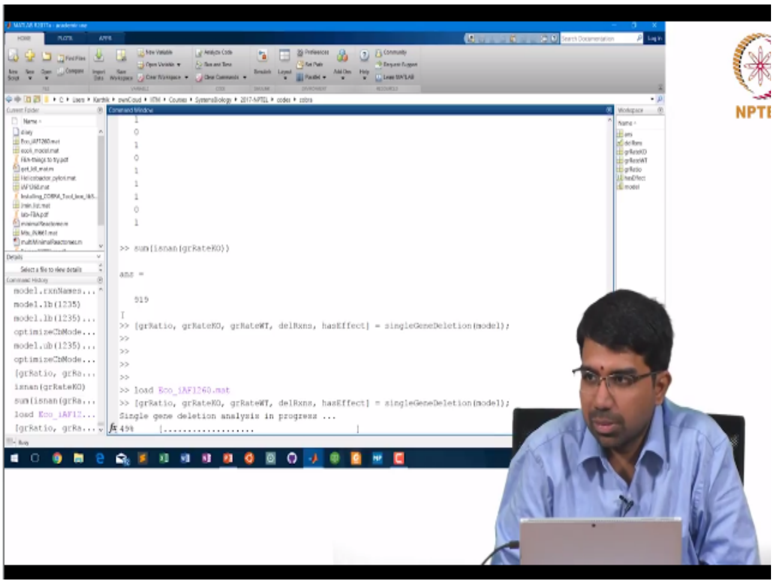


```
sum(isnan(grRa...
find(isnan(grR...
model.rxnNames...
model.lb(1235)
model.lb(1235)...
optimizeCbMode...
model.ub(1235)...
optimizeCbMode...
[grRatio, grRa...
isnan(grRateKO)
sum(isnan(grRa...
```

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

>> sum(isnan(grRateKO))

ans =

   919

fx >>
```

There is several gene deletions that give you nans. We have to optimise the model and see whether that gene deletion is (()) (10:00). That happen inside of single gene deletion. I will just do a simple analysis that should help. So let us now say, wait maybe, okay. I am just going to compare between setting ATP to 0 and keeping ATP as is and see how different are the results.

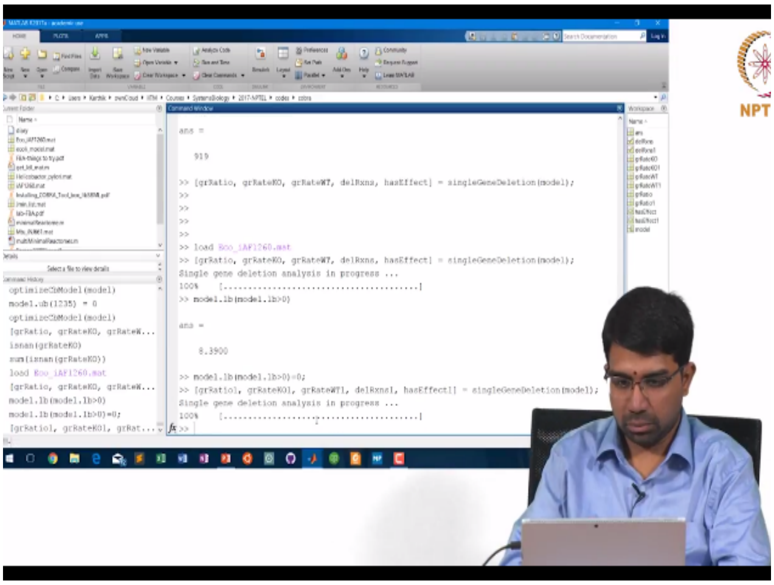
(Refer Slide Time: 10:45)



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

>> sum(isnan(grRateKO))

ans =

   919

>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleGeneDeletion(model);
>>
>>
>>
>>
>> load Eco_iAF1260.mat
>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleGeneDeletion(model);
Single gene deletion analysis in progress ...
49%     [...................                    ]
```

What are the changes and any of them are infeasible? Okay so this (()) (11:18) difference will give you. This growth rate knockout difference I am going to check. How many of them are like nans? There are many not a numbers that are showing up. In the 8.39 only, well I do not know, I have not done the other one. So in the 8.391, yes, I do not know how many showed up because I had changed one more constraint as well, this, let me just check.

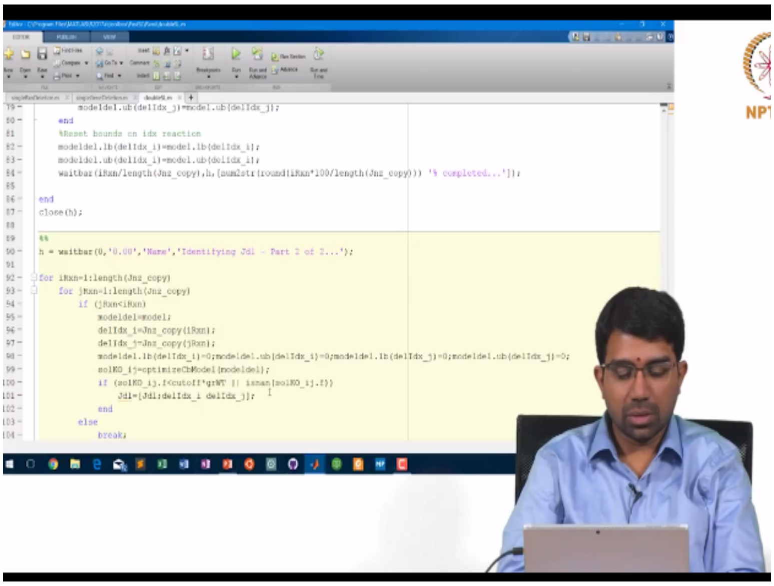
(Refer Slide Time: 11:45)



```
ans =

   919

>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleGeneDeletion(model);
>>
>>
>>
>>
>> load Eco_iAF1260.mat
>> [grRatio, grRateKO, grRateWT, delRxns, hasEffect] = singleGeneDeletion(model);
Single gene deletion analysis in progress ...
100%    [........................................]
>> model.lb(model.lb>0)

ans =

    8.3900

>> model.lb(model.lb>0)=0;
>> [grRatio1, grRateKO1, grRateWT1, delRxns1, hasEffect1] = singleGeneDeletion(model);
Single gene deletion analysis in progress ...
100%    [........................................]
>>
```

There was nothing, so I am going to how to find out deletions that break ATP. So there is no nan when ld are 0 when there is nan at ld is 8.39 is what that mean? No, no, no in both cases I got no nans. Otherwise yes, otherwise there will be exactly what we want. It turns out that here, but if you look at this is faster cell code and you look at the criteria.

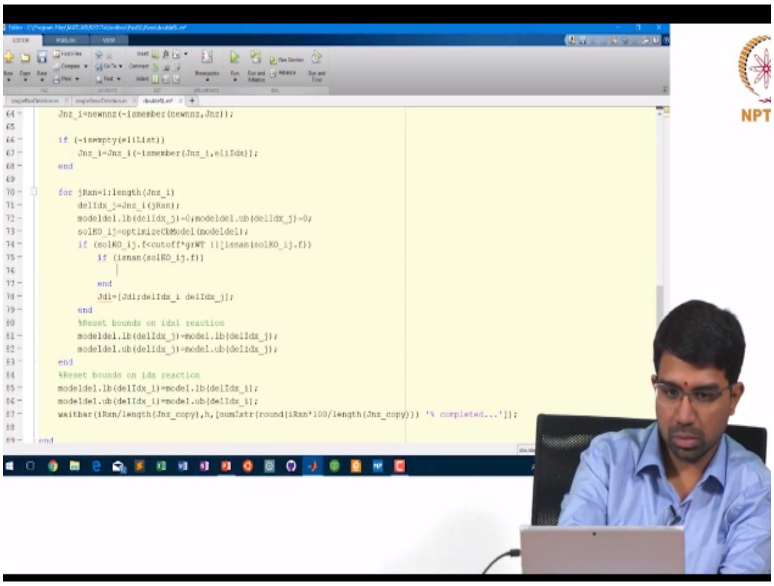
(Refer Slide Time: 12:45)



```
 79     modeldel.ub(delIdx_j)=model.ub(delIdx_j);
 80   end
 81   %Reset bounds on idx reaction
 82   modeldel.lb(delIdx_i)=model.lb(delIdx_i);
 83   modeldel.ub(delIdx_i)=model.ub(delIdx_i);
 84   waitbar(iRxn/length(Jnz_copy),h,[num2str(round(iRxn*100/length(Jnz_copy))) '% completed...']);
 85
 86 end
 87 close(h);
 88
 89 %%
 90 h = waitbar(0,'0.00','Name','Identifying Jdl - Part 2 of 2...');
 91
 92 for iRxn=1:length(Jnz_copy)
 93     for jRxn=1:length(Jnz_copy)
 94         if (jRxn<iRxn)
 95             modeldel=model;
 96             delIdx_i=Jnz_copy(iRxn);
 97             delIdx_j=Jnz_copy(jRxn);
 98             modeldel.lb(delIdx_i)=0;modeldel.ub(delIdx_i)=0;modeldel.lb(delIdx_j)=0;modeldel.ub(delIdx_j)=0;
 99             solKO_ij=optimizeCbModel(modeldel);
100             if (solKO_ij.f<cutoff*grWT || isnan(solKO_ij.f))
101                 Jdl=[Jdl;delIdx_i delIdx_j];
102             end
103         else
104             break;
```

So we are adding line 100, we are saying if this is solution of knockout dot f is less than cut off into growth rate of wild type which is like 0.05 or it is nan we are still adding it to JDL list of double gene deletions (()) (13:05) we talked about it.

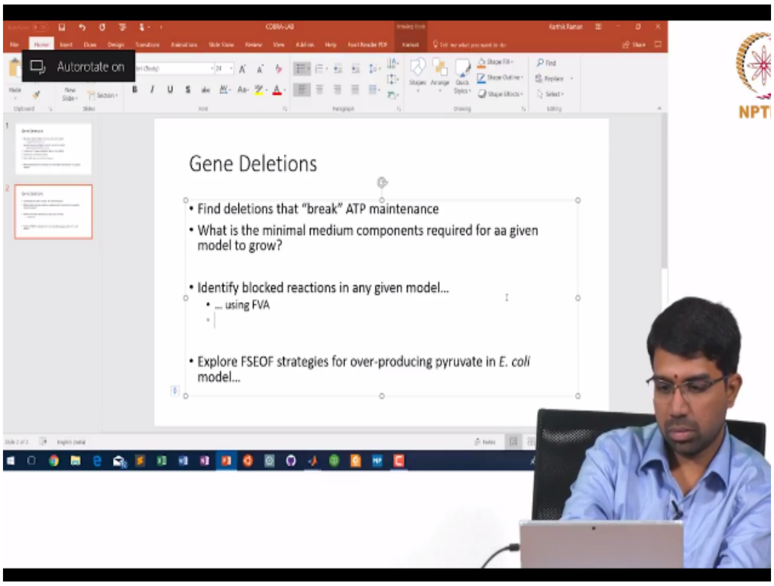
(Refer Slide Time: 13:45)



```
    Jnz_i=newnnz(~ismember(newnnz,Jnz));

    if (~isempty(eliList))
        Jnz_i=Jnz_i(~ismember(Jnz_i,eliIdx));
    end

    for jRxn=1:length(Jnz_i)
        delIdx_j=Jnz_i(jRxn);
        modeldel.lb(delIdx_j)=0;modeldel.ub(delIdx_j)=0;
        solKO_ij=optimizeCbModel(modeldel);
        if (solKO_ij.f<cutoff*grWT ||isnan(solKO_ij.f))
            if (isnan(solKO_ij.f))

            end
            Jdl=[Jdl;delIdx_i delIdx_j];
        end
        %Reset bounds on idx1 reaction
        modeldel.lb(delIdx_j)=model.lb(delIdx_j);
        modeldel.ub(delIdx_j)=model.ub(delIdx_j);
    end
    %Reset bounds on idx reaction
    modeldel.lb(delIdx_i)=model.lb(delIdx_i);
    modeldel.ub(delIdx_i)=model.ub(delIdx_i);
    waitbar(iRxn/length(Jnz_copy),h,[num2str(round(iRxn*100/length(Jnz_copy))) '% completed...']);

end
```

Also try to identify blocked reactions.

(Refer Slide Time: 14:45)

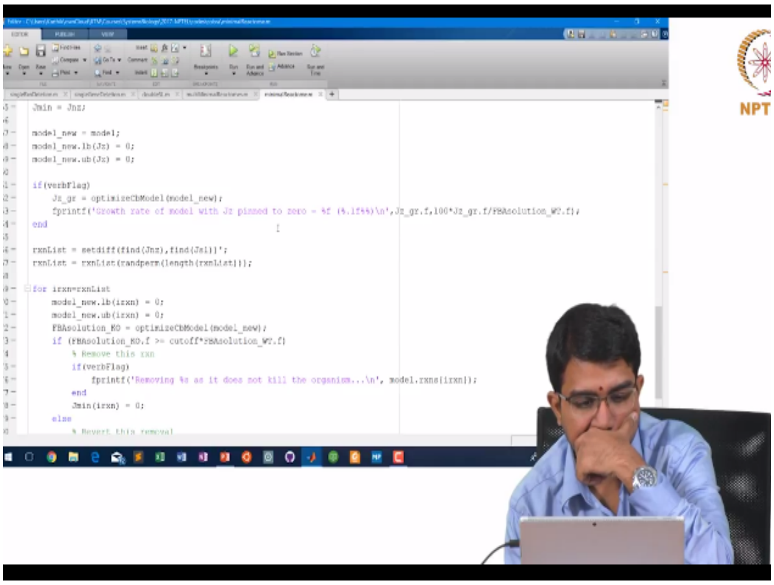


Gene Deletions

- Find deletions that "break" ATP maintenance
- What is the minimal medium components required for aa given model to grow?
- Identify blocked reactions in any given model…
  - … using FVA
- Explore FSEOF strategies for over-producing pyruvate in *E. coli* model…

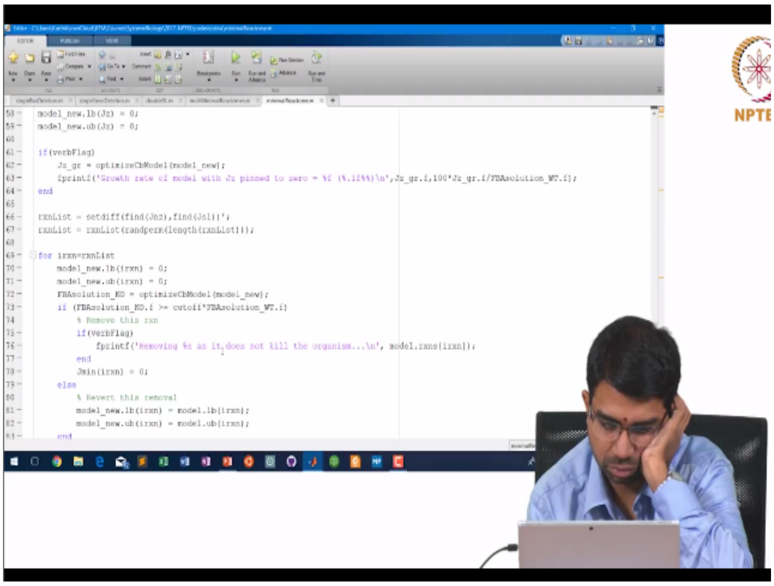
So far I have not found any example where there is nan. We have been doing double gene deletions. Nan means there is no flux or? No it is basically storing it as nan whenever the solver fails and that is exactly what we want.

(Refer Slide Time: 15:45)



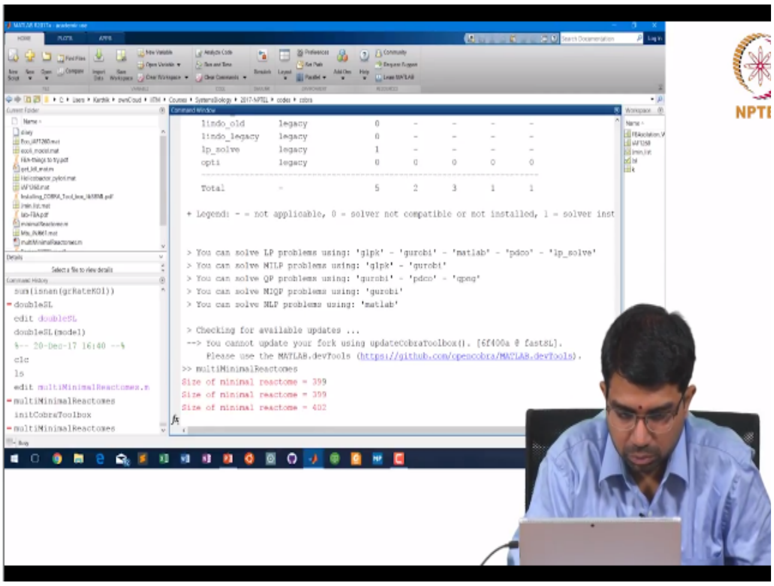
```
Jmin = Jnz;

model_new = model;
model_new.lb(Jz) = 0;
model_new.ub(Jz) = 0;

if(verbFlag)
    Jz_gr = optimizeCbModel(model_new);
    fprintf('Growth rate of model with Jz pinned to zero = %f (%.1f%%)\n',Jz_gr.f,100*Jz_gr.f/FBAsolution_WT.f);
end

rxnList = setdiff(find(Jnz),find(Jsl))';
rxnList = rxnList(randperm(length(rxnList)));

for irxn=rxnList
    model_new.lb(irxn) = 0;
    model_new.ub(irxn) = 0;
    FBAsolution_KO = optimizeCbModel(model_new);
    if (FBAsolution_KO.f >= cutoff*FBAsolution_WT.f)
        % Remove this rxn
        if(verbFlag)
            fprintf('Removing %s as it does not kill the organism...\n', model.rxns{irxn});
        end
        Jmin(irxn) = 0;
    else
        % Revert this removal
```

(Refer Slide Time: 16:45)



```
model_new.lb(Jz) = 0;
model_new.ub(Jz) = 0;

if(verbFlag)
    Jz_gr = optimizeCbModel(model_new);
    fprintf('Growth rate of model with Jz pinned to zero = %f (%.1f%%)\n',Jz_gr.f,100*Jz_gr.f/FBAsolution_WT.f);
end

rxnList = setdiff(find(Jnz),find(Jsl))';
rxnList = rxnList(randperm(length(rxnList)));

for irxn=rxnList
    model_new.lb(irxn) = 0;
    model_new.ub(irxn) = 0;
    FBAsolution_KO = optimizeCbModel(model_new);
    if (FBAsolution_KO.f >= cutoff*FBAsolution_WT.f)
        % Remove this rxn
        if(verbFlag)
            fprintf('Removing %s as it does not kill the organism...\n', model.rxns(irxn));
        end
        Jmin(irxn) = 0;
    else
        % Revert this removal
        model_new.lb(irxn) = model.lb(irxn);
        model_new.ub(irxn) = model.ub(irxn);
    end
```

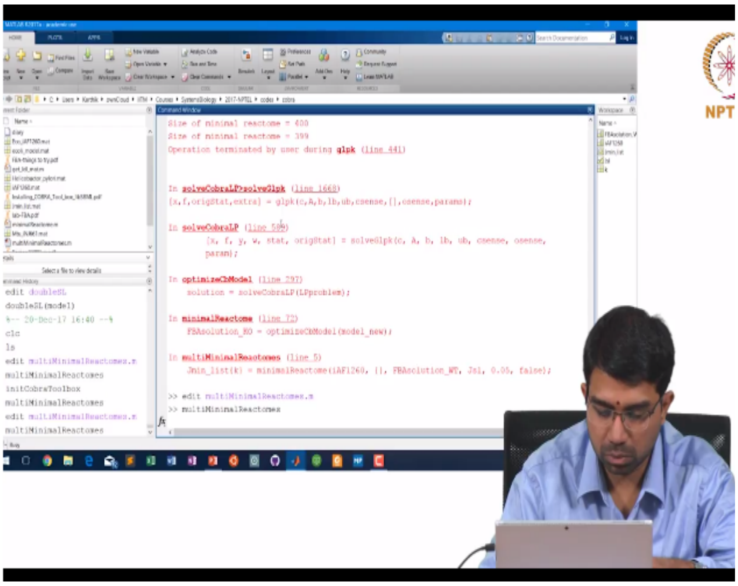
(Refer Slide Time: 17:45)



```
sum(isnan(grRateKO1))
doubleSL
edit doubleSL
doubleSL(model)
%-- 20-Dec-17 16:40 --%
clc
ls
edit multiMinimalReactomes.m
multiMinimalReactomes
initCobraToolbox
multiMinimalReactomes
```

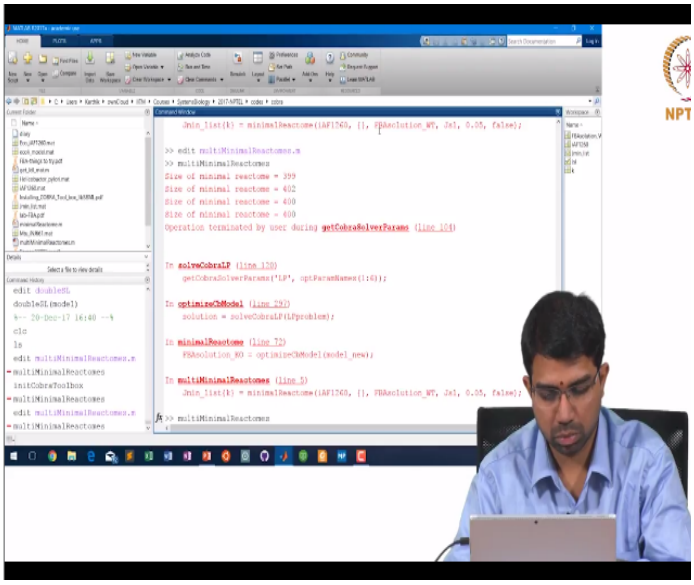
```
    lindo_old        legacy           0      -      -      -      -
    lindo_legacy     legacy           0      -      -      -      -
    lp_solve         legacy           1      -      -      -      -
    opti             legacy           0      0      0      0      0
    ---------------------------------------------------------------
    Total            -                5      2      3      1      1

 + Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver inst


 > You can solve LP problems using: 'glpk' - 'gurobi' - 'matlab' - 'pdco' - 'lp_solve'
 > You can solve MILP problems using: 'glpk' - 'gurobi'
 > You can solve QP problems using: 'gurobi' - 'pdco' - 'qpng'
 > You can solve MIQP problems using: 'gurobi'
 > You can solve NLP problems using: 'matlab'

 > Checking for available updates ...
 --> You cannot update your fork using updateCobraToolbox(). [6f400a @ fastSL].
     Please use the MATLAB.devTools (https://github.com/opencobra/MATLAB.devTools).
>> multiMinimalReactomes
Size of minimal reactome = 399
Size of minimal reactome = 399
Size of minimal reactome = 402
```

(Refer Slide Time: 18:45)



```
Size of minimal reactome = 400
Size of minimal reactome = 399
Operation terminated by user during glpk (line 441)


In solveCobraLP>solveGlpk (line 1668)
[x,f,origStat,extra] = glpk(c,A,b,lb,ub,csense,[],osense,params);

In solveCobraLP (line 599)
      [x, f, y, w, stat, origStat] = solveGlpk(c, A, b, lb, ub, csense, osense,
      param);

In optimizeCbModel (line 297)
  solution = solveCobraLP(LPproblem);

In minimalReactome (line 72)
  FBAsolution_KO = optimizeCbModel(model_new);

In multiMinimalReactomes (line 5)
  Jmin_list(k) = minimalReactome(iAF1260, [], FBAsolution_WT, Jsl, 0.05, false);

>> edit multiMinimalReactomes.m
>> multiMinimalReactomes
```

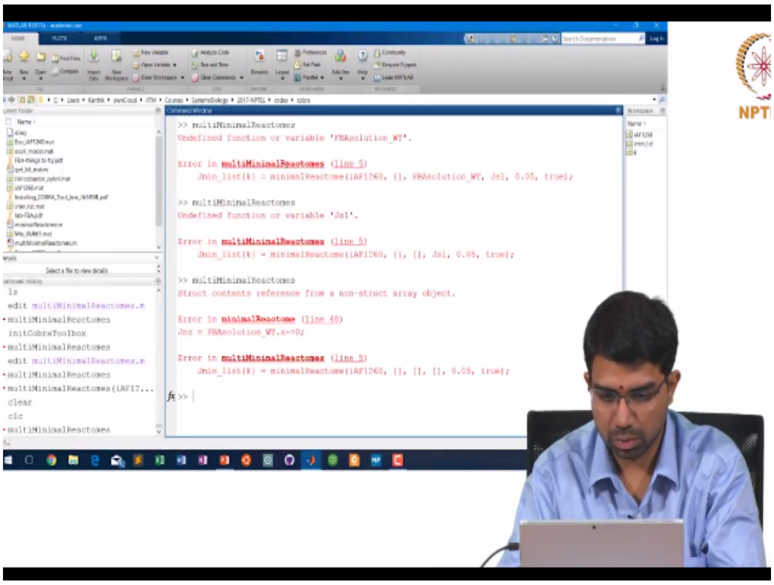
(Refer Slide Time: 19:45)



```
  Jmin_list(k) = minimalReactome(iAF1260, [], FBAsolution_WT, Jsl, 0.05, false);

>> edit multiMinimalReactomes.m
>> multiMinimalReactomes
Size of minimal reactome = 399
Size of minimal reactome = 402
Size of minimal reactome = 400
Size of minimal reactome = 400
Operation terminated by user during getCobraSolverParams (line 104)


In solveCobraLP (line 120)
  getCobraSolverParams('LP', optParamNames(1:6));

In optimizeCbModel (line 297)
  solution = solveCobraLP(LPproblem);

In minimalReactome (line 72)
  FBAsolution_KO = optimizeCbModel(model_new);

In multiMinimalReactomes (line 5)
  Jmin_list(k) = minimalReactome(iAF1260, [], FBAsolution_WT, Jsl, 0.05, false);

>> multiMinimalReactomes
```

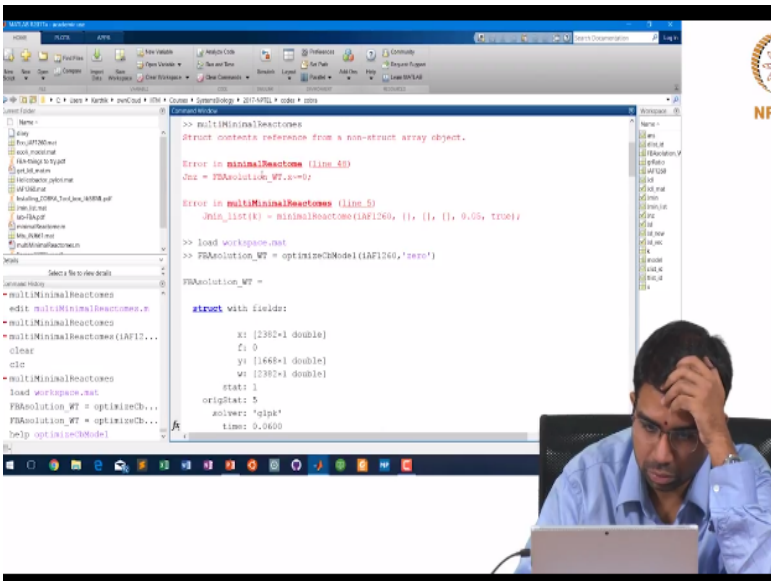
Sir delete model genes, should we give gene names as list or how? Gene names **"Professor - student conversation ends"**

(Refer Slide Time: 20:45)



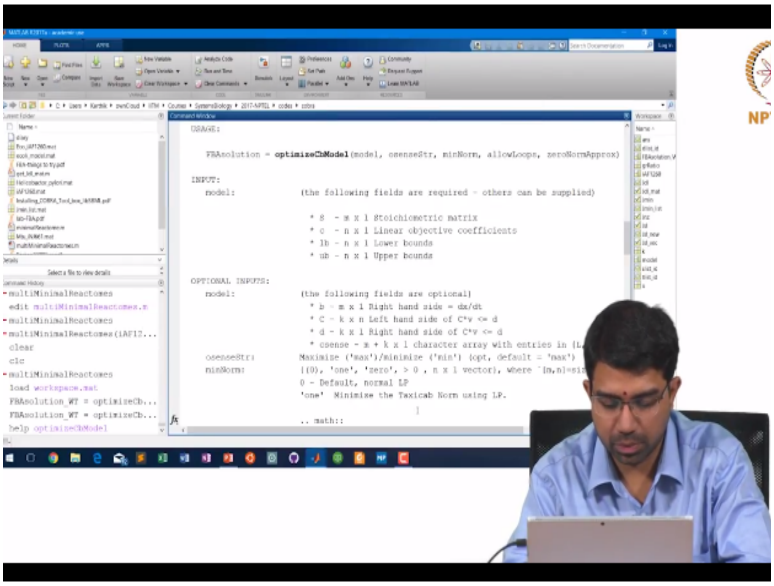
```
>> multiMinimalReactomes
Undefined function or variable 'FBAsolution_WT'.

Error in multiMinimalReactomes (line 5)
    Jmin_list(k) = minimalReactome(iAF1260, [], FBAsolution_WT, Jsl, 0.05, true);

>> multiMinimalReactomes
Undefined function or variable 'Jsl'.

Error in multiMinimalReactomes (line 5)
    Jmin_list(k) = minimalReactome(iAF1260, [], [], Jsl, 0.05, true);

>> multiMinimalReactomes
Struct contents reference from a non-struct array object.

Error in minimalReactome (line 48)
Jnz = FBAsolution_WT.x~=0;

Error in multiMinimalReactomes (line 5)
    Jmin_list(k) = minimalReactome(iAF1260, [], [], [], 0.05, true);

>> 
```

(Refer Slide Time: 21:45)



```
>> multiMinimalReactomes
Struct contents reference from a non-struct array object.

Error in minimalReactome (line 48)
Jnz = FBAsolution_WT.x~=0;

Error in multiMinimalReactomes (line 5)
    Jmin_list(k) = minimalReactome(iAF1260, [], [], [], 0.05, true);

>> load workspace.mat
>> FBAsolution_WT = optimizeCbModel(iAF1260,'zero')

FBAsolution_WT = 

  struct with fields:

           x: [2382×1 double]
           f: 0
           y: [1668×1 double]
           w: [2382×1 double]
        stat: 1
    origStat: 5
      solver: 'glpk'
        time: 0.0600
```

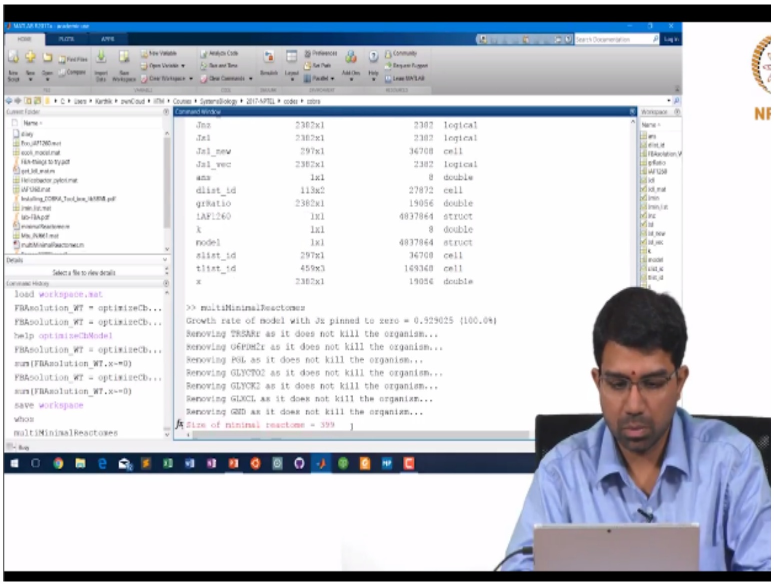
(Refer Slide Time: 22:45)



```
multiMinimalReactomes
edit multiMinimalReactomes.m
multiMinimalReactomes
multiMinimalReactomes(iAF12...
clear
clc
multiMinimalReactomes
load workspace.mat
FBAsolution_WT = optimizeCb...
FBAsolution_WT = optimizeCb...
help optimizeCbModel
```

```
USAGE:

   FBAsolution = optimizeCbModel(model, osenseStr, minNorm, allowLoops, zeroNormApprox)

INPUT:
   model:            (the following fields are required - others can be supplied)

                       * S  - m x 1 Stoichiometric matrix
                       * c  - n x 1 Linear objective coefficients
                       * lb - n x 1 Lower bounds
                       * ub - n x 1 Upper bounds

OPTIONAL INPUTS:
   model:            (the following fields are optional)
                       * b - m x 1 Right hand side = dx/dt
                       * C - k x n Left hand side of C*v <= d
                       * d - k x 1 Right hand side of C*v <= d
                       * csense - m + k x 1 character array with entries in {L,
   osenseStr:        Maximize ('max')/minimize ('min') (opt, default = 'max')
   minNorm:          {(0), 'one', 'zero', > 0 , n x 1 vector}, where `[m,n]=si
                     0 - Default, normal LP
                     'one'  Minimize the Taxicab Norm using LP.

                     .. math::
```

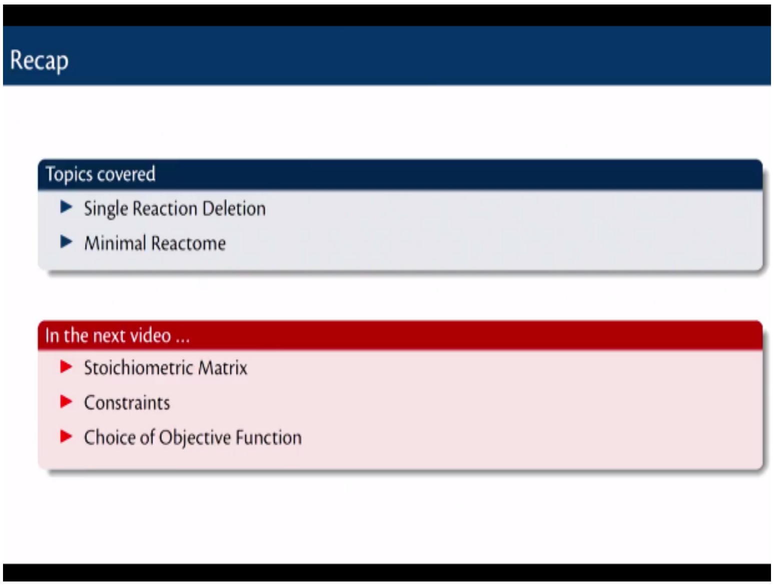
(Refer Slide Time: 23:45)



```
load workspace.mat
FBAsolution_WT = optimizeCb...
FBAsolution_WT = optimizeCb...
help optimizeCbModel
FBAsolution_WT = optimizeCb...
sum(FBAsolution_WT.x~=0)
FBAsolution_WT = optimizeCb...
sum(FBAsolution_WT.x~=0)
save workspace
whos
multiMinimalReactomes
```

```
  Jnz              2382x1                  2382  logical
  Jsl              2382x1                  2382  logical
  Jsl_new           297x1                 36708  cell
  Jsl_vec          2382x1                  2382  logical
  ans                 1x1                     8  double
  dlist_id          113x2                 27872  cell
  grRatio          2382x1                 19056  double
  iAF1260             1x1               4837864  struct
  k                   1x1                     8  double
  model               1x1               4837864  struct
  slist_id          297x1                 36708  cell
  tlist_id          459x3                169368  cell
  x                2382x1                 19056  double

>> multiMinimalReactomes
Growth rate of model with Jz pinned to zero = 0.929025 (100.0%)
Removing TRSARr as it does not kill the organism...
Removing G6PDH2r as it does not kill the organism...
Removing PGL as it does not kill the organism...
Removing GLYCTO2 as it does not kill the organism...
Removing GLYCK2 as it does not kill the organism...
Removing GLXCL as it does not kill the organism...
Removing GND as it does not kill the organism...
Size of minimal reactome = 399
```

(Refer Slide Time: 24:35)



Recap

Topics covered

- Single Reaction Deletion
- Minimal Reactome

In the next video ...

- Stoichiometric Matrix
- Constraints
- Choice of Objective Function

So in today's lab session, we looked at single reaction deletion and how we can try to assemble the minimal reactome for an organism and in the next video, I will conclude constraint based modeling by recapping all the concepts we have covered so far beginning with the stoichiometric matrix, the kind of constraints that we impose and again the favourite topic which is the choice of objective function for flux balance analysis.