MetaboRePort Tutorial

```
% Set path to the cobratoolbox
global CBTDIR

% Set root directory
root = '';

% user defined path
folder = [root filesep 'refinedReconstructions']; % Set path to folder with
reconstructions

% Make folder to save the updated reconstructions
updatedReconstructPath = [root filesep 'updatedReconstructions'];
if exist(updatedReconstructPath,'dir')~=7
    mkdir(updatedReconstructPath)
end

% Make folder to save the updated reconstructions as sbml files
annotatedSBMLreconstructions = [root filesep 'annotatedSBMLreconstructions'];
if exist(annotatedSBMLreconstructions,'dir')~=7
    mkdir(annotatedSBMLreconstructions)
end

% Make folder to where the reports are saved
reportDir = [root filesep 'reports'];
if exist(reportDir,'dir')~=7
    mkdir(reportDir)
end

% Load rBioNet metabolite structure information
metstructPath = [CBTDIR filesep 'tutorials' filesep 'dataIntegration'
filesep...
 'metaboAnnotator' filesep 'data' filesep 'met_strc_rBioNet_new.mat'];

% Ensure that the name of the rBioNet metabolite structure is
metabolite_structure_rBioNet
metabolite_structure_rBioNet = load(metstructPath);
metabolite_structure_rBioNet = metabolite_structure_rBioNet.
(string(fieldnames(metabolite_structure_rBioNet)));

% Get reconstructions and reconstruction paths
directory = what(folder);
modelPaths = append(directory.path, filesep, directory.mat);
modelList = getModelPaths(folder);

% Preallocate ScoresOverall table for speed
ScoresOverall = cell(length(modelList),2);

tic;
```

```
for i = 1 : length(modelList)
    disp(i)
    % Load model
    model = load(modelPaths(i));
    model = model.(string(fieldnames(model))); % ensure that the name of the
loaded model is "model".

    %[modelProp1,ScoresOverall1] = generateMemoteLikeScore(model);
    % Populate and further annotate model with metabolite info
    [modelUpdated] = populateModelMetStr(model,
metabolite_structure_rBioNet,1);
    [modelUpdated] = annotateSBOTerms(modelUpdated);

    if any(contains(fieldnames(modelUpdated), {'metInChIString'}))
        modelUpdated = rmfield(modelUpdated,'metInChIString'); % wrongly in
microbe models
    end

    [modelUpdated] = populateModelwithRxnIDs(modelUpdated);

    [modelProp2,ScoresOverall2] = generateMetaboScore(modelUpdated);

    modelProperties.(regexprep(modelList{i},'.mat','')).ScoresOverall =
ScoresOverall2;
    modelProperties.(regexprep(modelList{i},'.mat','')).modelUpdated =
modelUpdated;
    modelProperties.(regexprep(modelList{i},'.mat','')).modelProp2 =
modelProp2;
    ScoresOverall{i,1} = regexprep(modelList{i},'.mat','');
    ScoresOverall{i,2} = num2str(ScoresOverall2);

    if mod(i,10) % Save every ten models
        save('MetaboRePorts.mat','modelProperties','ScoresOverall');
    end

    % save updated mat file
    model = modelUpdated;
    save(strcat(updatedReconstructPath, filesep, modelList(i),
'.mat'),'model');

    % generate sbml file
    % remove description from model structure as this causes issues
    if any(contains(fieldnames(modelUpdated), {'description'}))
        modelUpdated = rmfield(modelUpdated,'description');
    end

    % Set sbml path
    sbmlPath = char(strcat(annotatedSBMLreconstructions, filesep,
'Annotated_',modelList(i)));
    % Save model
    outmodel = writeCbModel(modelUpdated, 'format','sbml', 'fileName',
sbmlPath);
end
toc;

% Generate a generateMetaboReport for each reconstruction
evalc('generateMetaboReport(modelProperties,reportDir)');
```