TN-CORE: CONTEXT-SPECIFIC RECONSTRUCTION OF CORE METABOLIC MODELS USING TN-SEQ DATA

George C diCenzo, Alessio Mengoni, Marco Fondi

1. TABLE OF CONTENTS

2. Overview

Tn-Core was developed with the intent of facilitating the generation of context-specific core metabolic models through the integration Tn-seq data, as well as through the combined integration of both Tn-seq and RNA-seq data. Tn-Core is further designed to assist in the analysis of redundancy in core metabolic networks, and the identification of biochemical pathways contributing to improved cellular growth.

At the time of publication, Tn-Core consists of eight scripts. These scripts are written in Matlab and make use of Cobra formatted models and the cobraToolbox. The most recent version of Tn-Core is available in the following GitHub repository: https://github.com/diCenzo-GC/Tn-Core.

A more detailed description of the functionality of Tn-Core can be found in the associated publication that describes its development, available here:
diCenzo GC, Mengoni A, Fondi M. 2017. Tn-Core: context-specific reconstruction of core metabolic models using Tn-seq data. *bioRxiv*. doi: 10.1101/221325

3. Citation

If you make use of any of the scripts of Tn-Core or any of its output files, please cite the following article:

diCenzo GC, Mengoni A, Fondi M. 2017. Tn-Core: context-specific reconstruction of core metabolic models using Tn-seq data. *bioRxiv*. doi: 10.1101/221325

4. Contact information

If you have question related to the usage of Tn-Core, please contact either George diCenzo at georgecolin.dicenzo@unifi.it and/or Marco Fondi at marco.fondi@unifi.it.

4. SCRIPT: TNCORE_MAIN

Description

Tncore_main is the primary script of the Tn-Core toolbox. Use this script to generate a population of randomized core metabolic models, with or without the presence of Tn-seq data. If Tn-seq data is included, the script will return two Cobra formatted core models, the one most consistent with the Tn-seq data, and the one with the highest objective flux. Regardless of if Tn-seq data is provided, several files will be returned for use with the *Tncore_matrix* script to generate summary matrixes representing the variation and redundancy within all core model that were produced.

Usage

[coreGeneVar, coreRxnVar, coreGrowthVar, coreModel, coreModelFast, reducedModel, … coreGeneCat, genePresence, rxnPresence] = tncore_main(model, iters, growthThresh, ... tnseq, coreGenes, rnaseq, expressThresh, method)

Inputs

model: A Cobra-formatted, genome-scale metabolic network reconstruction from which the core metabolic models are to be derived. The model must include gene-reaction associations. Exchange bounds must be set to represent the desired growth environment and nutrient uptake rate prior to the use of the model in this script.

Optional inputs

iters: The number of random core metabolic models to be produced. If no value is provided, the default value is 1,000. However, in most cases this will be insufficient, and we recommend using a minimum of 20,000 iterations, if not more.

growthThresh: The minimum allowable objective flux in the generated core models. If no value is provided, the default threshold is set to 10% the objective flux of the input model.

tnseq: The Tn-seq data. Data should be provided for all genes in the organism, not only the genes in the model, in order to allow correct essentiality thresholds to be determined. This variable should be a matrix, with the first column containing the Tn-seq data (not log transformed), and the second column containing the gene names. If no Tn-seq data is provided, the *coreModel* and *coreModelFast* outputs are not returned.

coreGenes: A list of genes to be protected (i.e., not directly deleted) during generation of the core metabolic models. Note that a gene in this list may still be removed from the model if its associated reaction is constrained due to the deletion of another gene. This field can either be empty (no core genes are set prior to core model generation), contain a list of genes (genes to be protected), or set to {1} (core genes are automatically determined by

the script on the basis of the Tn-seq data as those with a log-transformed value < the median - 3.5 * the standard deviation).

rnaseq: The RNA-seq data. Data should be provided for all genes in the organism, not only the genes in the model, in order to allow correct expression thresholds to be determined if not set by the user. This variable should be a matrix, with the first column containing the RNA-seq data (not log transformed), and the second column containing the gene names. Genes above the expression threshold are included in the core gene list, and protected during core model generation; however, a highly expressed gene may still be removed from the model if its associated reaction is constrained due to the deletion of another gene. RNA-seq data can only be provided if Tn-seq data is also provided.

expressThresh: The threshold for a gene to be considered expressed. If no value is provided, the default is set to 0.02% the sum of all expression values.

method: Set the script to use either the FBA or MOMA algorithms when calculating growth rates of gene deletion mutants. Use 1 for FBA and 2 for MOMA (Default = 1)

Outputs

coreGeneVar: A matrix where each column represents one core model, and each row represents a gene from the reduced input model. If the gene was included as part of the core model, the gene name is given. If the gene was not included as part of the core model, the gene is given but appended with '_deleted'.

coreRxnVar: A matrix where each column represents one core model, and each row represents a reaction from the reduced input model. If the reaction was included as part of the core model, the reaction ID is given. If the reaction was not included as part of the core model, the reaction ID is given but appended with '_deleted'. Models (i.e., columns) are in the same order as those in the *coreGeneVar* output.

coreGrowthVar: A matrix consisting of one row, with each column representing one core model. The objective flux rate for each of the core models is indicated. The values are provided in the same order as the models in the *coreGeneVar* output; i.e., the objective flux in the first cell corresponds to the model of the first column of the *coreGeneVar* variable.

coreModel: The Cobra-formatted core model derived from the input model that was most consistent with the Tn-seq data. This output is not provided if no Tn-seq data is given as input.

coreModelFast: The Cobra-formatted core model derived from the input model that had the highest rate of flux through the objective reaction. This output is not provided if no Tn-seq data is given as input.

reducedModel: A reduced version of the input model. This model is the input model, but with all reactions involving dead-ends metabolites removed from the model, and with all 'or Unknown' GPRs removed.

coreGeneCat: Contains four rows corresponding to the different gene fitness categories, with each column corresponding to one model in the same order as the *coreGeneVar* variable. Each cell contains the number of genes in the model that was grouped into that fitness category based on the Tn-seq data. The top row are non-essential genes, the second row are genes whose deletion results in a moderate growth impairment, the third row are genes whose deletion results in a severe growth impairment, and the fourth row are the essential genes.

genePresence: The same as *coreGeneVar* except that gene presence is indicated by a '1' and gene absence is indicated by a '0'.

rxnPresence: The same as *coreRxnVar* except that reaction presence is indicated by a '1' and reaction absence is indicated by a '0'.

Notes

When many iterations are run, the output files *coreGeneVar*, *coreRxnVar*, *genePresence*, and *rxnPresence* are too big to be saved as a standard '.mat' file. In this case, it is recommended to convert the cell arrays to tables, and to export the tables as text files.

If you wish to modify the threshold for what is classified as an essential gene when *coreGenes* is set to {1}, this can be modified at line 243 of this script.

If you wish to modify the thresholds for what the fitness grouping of genes are during the core model generation, this can be modified at lines 205-215 of the *tncore_randomize* script.

5. SCRIPT: TNCORE_RANDOMIZE

Description

Tncore_randomize is the script used to generate a randomized core metabolic model. It is called during the running of *tncore_main*, but can also be used as a stand alone script to produce a single core model (alternatively the *tncore_main* script can be run with an iteration of 1).

Usage

```
[coreModel, genesTnseq, geneGrouping, solution, reactions, rxnPresence, genePresence] = …
    tncore_randomize(model, growthThresh, nonProtected, modelTnseq, medianValue, …
    stdev, solver, method)
```

Inputs

model: A Cobra-formatted, genome-scale metabolic network reconstruction from which the core metabolic models are to be derived. The model must include gene-reaction associations. Exchange bounds must be set to represent the desired growth environment and nutrient uptake rate prior to the use of the model in this script.

Optional inputs

growthThresh: The minimum allowable objective flux in the generated core models. If no value is provided, the default threshold is set to 10% the objective flux of the input model.

nonProtected: A list of genes that can be the script can attempt to delete during construction of the core metabolic model. If no genes are provided, a default list is prepared consisting of genes whose deletion results in an objective flux below the growth threshold.

modelTnseq: The Tn-seq data. Data should be provided only for the genes present in the model. This variable should be a matrix, with the first column containing the log-transformed Tn-seq data, and the second column containing the gene names.

medianValue: The median of the log-transformed Tn-seq data for all genes in the genome, following the removal of outlier values. If no Tn-seq data is provided, this field is left empty.

stdev: The standard deviation of the log-transformed Tn-seq data for all genes in the genome, following the removal of outlier values. If no Tn-seq data is provided, this field is left empty.

solver: The Cobra LP solver to be used. If no input is given, the default is set to the currently set Cobra LP solver.

method: Set the script to use either the FBA or MOMA algorithms when calculating growth rates of gene deletion mutants. Use 1 for FBA and 2 for MOMA (Default = 1)

Outputs

coreModel: The Cobra-formatted core model produced from the input model.

geneTnseq: A list of the Tn-seq values for the genes included in the core model. Values are provided in the order of genes in the input model genes field.

geneGrouping: Contains four cells corresponding to the different gene fitness categories. Each cell contains the number of genes in the model that was grouped into that fitness category based on the Tn-seq data. The top row are non-essential genes, the second row are genes whose deletion results in a moderate growth impairment, the third row are genes whose deletion results in a severe growth impairment, and the fourth row are the essential genes.

solution: The objective flux rate for the core models is indicated, determined with the *optimizeCbModel* function of the cobraToolbox.

reactions: A list indicating which reactions were included in the core model. If the reaction was included as part of the core model, the reaction ID is given. If the reaction was not included as part of the core model, the reaction ID is given but appended with '_deleted'.

rxnPresence: A binary matrix indicating if each reaction present in the input model was included in the core model (1) or not included in the core model (0). Numbers correspond to the reaction in the same position of the input models rxn field.

genePresence: A binary matrix indicating if each gene present in the input model was included in the core model (1) or not included in the core model (0). Numbers correspond to the gene in the same position of the input models genesfield.

Notes

If you wish to modify the thresholds for what the fitness grouping of genes are during the core model generation, this can be modified at lines 205-215 of this script.

6. SCRIPT: TNCORE_MATRIX

Description

Tncore_matrix is used to generate a few matrixes to summarize the variation between the population of core models generated with the *tncore_main* script. The output matrixes can be directly used for the generation of figures.

Usage

[coocurMatrix, coocurMatrixAdj, growthMatrix, varPresenceLabel, uniqueModels] = tncore_matrix(model, variable, growth, varPresence, type, minPresence)

Inputs

model: The Cobra-formatted model from which the core metabolic models was generated. If the core models were produced with the *tncore_main* script, use the *reducedModel* output from the *tncore_main* script.

variable: The *coreGeneVar* or the *coreRxnVar* output variable from the *tncore_main* script.

growth: The *coreGrowthVar* output variable from the *tncore_main* script.

varPresence: The *genePresence* or *rxnPresence* output variable from the *tncore_main* script.

Optional inputs

type: The type of matrixes to produce. If gene matrixes are to be produced, use 1. If reaction matrixes are to be produced, use 2. Ensure that all input variables correspond to the correct type of matrixes to be produced. If no value in given, the default is 1.

minPresence: The minimum number of models a gene or reaction must be present in to be included within the output matrixes. If no value is given, the default is one model.

Outputs

coocurMatrix: A matrix indicating the total number of times each pair of genes or reactions occurred in the same core model. Each column and each row represents a gene, and each cell indicates the number of times those two genes were in the same model. Only genes or reactions included in at least as many models indicated in the *minPresence* input. The first row/column contains the gene name.

coocurMatrixAdj: A matrix that provides a Chi-squared statistic to indicate the significance of observed over- or under-representation of each pair of genes or reactions co-occurring in the same core model. Each column and each row represents a gene, and each cell contains a Chi-squared statistic indicating if the genes or reactions are more likely than expected to

occur in the same core model (a positive number) or more likely than expected to not occur in the same core model (a negative number). Only genes or reactions included in at least as many models indicated in the *minPresence* input. The first row/column contains the gene name.

growthMatrix: A matrix indicating the average growth rate of core models containing the indicated gene, the average growth rate of core models not containing the indicated gene, and the number of core models that the gene is included within.

varPresenceLabel: The *varPresence* input variable modified to row and column labels, and with genes or reactions that are not present in at least the number of models specified by the *minPresence* field removed.

uniqueModels: Indicates how many unique models were present in the core model population, based either on the genes present in the models (type = 1) or the reactions present in the models (type = 2).

7. SCRIPT: TNCORE_REFINE

Description

Tncore_refine is an extension of the *tncore_main* script that first generates a core model consistent with Tn-seq data, and then uses this core model to refine the GPRs of the input genome-scale metabolic model. In essence, the script first identifies the essential genes in the Tn-seq data, and generates a core model consistent with the Tn-seq data while protecting the essential genes. Once the core model is produced, any genes essential in the Tn-seq data but not the core model are identified. If two or more of these genes are associated with the same, and only the same, reaction(s), for any reaction that has only 'or' statements in the GPR, the 'or' statements are replaced with 'and' statements. At this point, for any reaction in the core model that contains an essential genes based on the Tn-seq data, the GPRs for these reactions in the input model are replaced with the corresponding GPRs of the core model.

Usage

[refinedModel] = tncore_refine(model, tnseq, iters, growthThresh, coreGenes, deadends)

Inputs

model: The Cobra-formatted, genome-scale metabolic network reconstruction that is to be refined through the integration of the Tn-seq data. The model must include gene-reaction associations. Exchange bounds must be set to represent the desired growth environment and nutrient uptake rate prior to the use of the model in this script.

tnseq: The Tn-seq data. Data should be provided for all genes in the organism, not only the genes in the model, in order to allow correct essentiality thresholds to be determined. This variable should be a matrix, with the first column containing the Tn-seq data (not log transformed), and the second column containing the gene names.

Optional inputs

iters: The number of random core metabolic models to be produced. If no value is provided, the default value is 1,000. However, in most cases this will be insufficient, and we recommend using a minimum of 20,000 iterations, if not more.

growthThresh: The minimum allowable objective flux in the generated core models. If no value is provided, the default threshold is set to 10% the objective flux of the input model.

coreGenes: A list of genes to be protected (i.e., not directly deleted) during generation of the core metabolic models. Note that a gene in this list may still be removed from the model if its associated reaction is constrained due to the deletion of another gene. This field can either be empty (no core genes are set prior to core model generation), contain a list of genes (genes to be protected), or set to {1} (core genes are automatically determined by

the script on the basis of the Tn-seq data as those with a log-transformed value < the median - 3.5 * the standard deviation).

deadends: To remove reactions producing dead-ends from the output, refined model, use 1. TO leave reactions producing dead-ends in the output, refined mode, use 0. If no value is provided, the default is to remove the reactions producing dead-ends.

Outputs

refinedModel: A Cobra-formatted model of the original, input genome-scale model, but with the GPRs associated with genes of the *coreGenes* list updated based on the GPRs of the core model, and optionally with dead-ends removed.

Notes

If you wish to modify the threshold for what is classified as an essential gene when *coreGenes* is set to {1}, this can be modified at line 101 of this script.

If you wish to modify the thresholds for what the fitness grouping of genes are during the core model generation, this can be modified at lines 205-215 of the *tncore_randomize* script.

8. SCRIPT: TNCORE_COMPARE

Description

Tncore_compare is meant to compare the frequency that genes or reactions occur in core model populations generated from independent runs of the *tncore_main* script. For example, if *tncore_main* is run multiple times using different growth thresholds, this script can take the output from the *tncore_main* runs, and indicate the frequency that each gene or each reaction is found in the core models for each growth threshold.

Usage

[varPresenceMatrix] = tncore_compare(varNames, varPresenceArray, headers)

Inputs

varNames: A cell array containing the gene names or the reaction IDs in the same order as the genes or reactions are given in the *genePresence* or *rxnPresence* variables.

varPresenceArray: A cell array containing the names of the *genePresence* or *rxnPresence* outputs from the multiple *tncore_main* runs. For example, if *tncore_main* was run three times, there should be three *genePresence* variables (e.g., genePres1, genePres2, genPres3). In this case, this variable should contain the following:
{'genePres1', 'genePres2', 'genePres3'}

Optional inputs

headers: A cell array containing the names to use as the column labels in the *varPresenceMatrix* output variable. If this is empty, the default is to use the *varPresenceArray* as the column labels.

Outputs

varPresenceMatrix: A matrix indicating the percent of core models in each core model population that contained the gene or reaction. Each row corresponds to a gene or reaction, and each column corresponds to a different set of core model populations. Column and row labels are included.

9. SCRIPT: TNCORE_RECONSTRUCT

Description

Tncore_reconstruct is designed to reproduce a core metabolic model based on a binary gene presence/absence list. Developed in order to allow reconstruction of any of the core models produced during random model core generation by the *tncore_main* script using the *reducedModel* output and one colume of the *genePresence* output table.

Usage

[coreModel] = tncore_reconstruct(model, genePresence, trim)

Inputs

model: The Cobra-formatted metabolic model from which the core model is derived. If reconstructing a core model generated by the *tncore_main* function, use the *reducedModel* exported by the *tncore_main* script.

genePresence: A binary array indicating which genes in the input model are presence (1) or absence (0) in the core model to be produced. Can use one column of the *genePresence* output of the *tncore_main* script.

Optional inputs.

trim: Indicates if non-essential reaction that are either non associated with any gene or have only an 'Uknown' GPR should be removed from the core model. Use 1 for trim, and 0 for do not trim. If no value is given, the default is to remove these reactions.

Outputs

coreModel: The core model produced from the input model and containing only the genes and associated reactions of the *genePresence* input.

11. SCRIPT: TNCORE_REMOVE

Description

Tncore_remove was written to remove genes in the gene list that are no longer found in the grRules field due to the deletion of reactions from the model. Genes are removed from the genes field, and the rules and rxnGeneMat fields are updated accordingly.

Usage

[modelNew] = tncore_remove(model)

Inputs

model: The model from which to remove the genes. The script will first identify which genes are no longer present in the grRules field, and then remove them from the model.

Outputs

modelNew: The input model, but with the genes not present in the grRules removed from the genesfield, and with the rules and rxnGeneMat fields updated accordingly.

12. SCRIPT: TNCORE_DELETE

Description

Tncore_delete is designed to remove any gene appended with '_deleted' in the genesfield (as is done using the *deleteModelGenes* script of the cobraToolbox) from the genesand grRules field, and to update the rules and rxnGeneMat fields accordingly. Note that it does not remove any reactions from the model. Can be useful following the combined use of *deleteModelGenes* and *removeRxns* of the cobraToolbox.

Usage

[modelNew] = tncore_delete(model)

Inputs

model: The model from which to delete the genes. At least one gene must be appended with '_deleted'.

Outputs

modelNew: The input model, but with the deleted genes removed from the genes and grRules fields, and with the rules and rxnGeneMat fields updated accordingly.