

Systems biology

Mackinac: a bridge between ModelSEED and COBRApy to generate and analyze genome-scale metabolic models

Michael Mundy^{1,*}, Helena Mendes-Soares^{1,3,*} and Nicholas Chia^{1,3,4}

¹Center for Individualized Medicine, ²Department of Information Technology, ³Department of Surgery and ⁴Department of Physiology and Biomedical Engineering, Mayo Clinic, Rochester, MN, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on February 8, 2017; revised on March 15, 2017; editorial decision on March 27, 2017; accepted on March 28, 2017

Abstract

Summary: Reconstructing and analyzing a large number of genome-scale metabolic models is a fundamental part of the integrated study of microbial communities; however, two of the most widely used frameworks for building and analyzing models use different metabolic network representations. Here we describe Mackinac, a Python package that combines ModelSEED's ability to automatically reconstruct metabolic models with COBRApy's advanced analysis capabilities to bridge the differences between the two frameworks and facilitate the study of the metabolic potential of microorganisms.

Availability and Implementation: This package works with Python 2.7, 3.4, and 3.5 on MacOS, Linux and Windows. The source code is available from <https://github.com/mmundy42/mackinac>.

Contact: mundy.michael@mayo.edu or soares.maria@mayo.edu

1 Introduction

Reconstructing genome-scale metabolic models (GEMs) is a complex process that involves integrating multiple data sources. A GEM for a particular organism can be reconstructed manually using a standard protocol and current knowledge from literature (Thiele and Palsson, 2010). Alternatively, a GEM can be reconstructed automatically, which enables the creation of models for the large number of organisms that typically make up microbial communities. Automated reconstruction uses the annotated genome of the organism to predict reactions to include in the draft GEM and specialized methods to gap fill missing reactions in the metabolic network (Benedict, 2014; Henry, 2010; Kumar and Maranas, 2009; Reed, 2006; Thiele, 2014).

ModelSEED is the most widely used of the existing frameworks for automated GEM reconstruction. Using the ModelSEED web service, a researcher can reconstruct and gap fill GEMs from a large database of reactions and functional roles. The GEMs can then be used to analyze the growth characteristics of the organisms and to evaluate the effects of reaction or gene knockouts using constraint-based analysis methods (Henry, 2010). COBRApy (Ebrahim, 2013) is

a Python package that uses constraint-based analysis to study the metabolism of both single organisms and microbial communities. This popular package is under continuous development, and new functionalities for model analysis and exploration are frequently added. While the ModelSEED web service and COBRApy are widely used in the field of microbial metabolism, they are independent frameworks. The integration of their capabilities to study organisms can only be done manually, which is a very laborious and time-consuming process if a large number of species is to be studied.

We developed Mackinac, a Python package that creates a COBRA model object directly from a ModelSEED model object, seamlessly providing a bridge between the two frameworks. The reconstruction of the ModelSEED model object is accomplished within the ModelSEED framework (Henry, 2010). The COBRA model then created contains all of the information from the ModelSEED model, including features that are commonly lost when the models are exported to the SBML format (Chaouiya, 2013) on the ModelSEED web service. Among these are the chemical equations of metabolites, and the names of the genes in the gene-protein-reaction evidence for a particular reaction. By allowing the

Table 1. List of additional Mackinac functions

Function	Description
<i>Model Functions</i>	
create_universal_model	Creates a universal model from a ModelSEED template model
delete_modelseed_model	Deletes a ModelSEED model from the workspace
get_modelseed_fba_solutions	Gets the list of flux balance analysis solutions available for a ModelSEED model
get_modelseed_gapfill_solutions	Gets the list of gap fill solutions available for a ModelSEED model
get_modelseed_model_data	Gets the model data for a ModelSEED model
get_modelseed_model_stats	Gets the model statistics for a ModelSEED model
list_modelseed_models	Lists the ModelSEED models
optimize_modelseed_model	Runs optimization of objective function
<i>Workspace Functions</i>	
delete_workspace_object	Deletes an object from the workspace
get_workspace_object_data	Gets the data for an object
get_workspace_object_meta	Gets the metadata for an object
list_workspace_objects	Lists the objects in the specified workspace folder
put_workspace_object	Puts an object and its metadata in the workspace
<i>Genome Functions</i>	
get_genome_features	Gets list of features from the annotation for a genome in PATRIC
get_genome_summary	Gets the summary data for a genome in PATRIC

reconstruction of models using the ModelSEED framework, Mackinac allows the comprehensive storage of all the information associated with the models in the COBRA model object, and provides direct access to many of the functions available from this web service, such as functions to reconstruct, gap fill and optimize GEMs. It also provides functions to manage and work with models stored in the user’s ModelSEED workspace. Thus, Mackinac combines ModelSEED’s ability to rapidly reconstruct GEMs with COBRApy’s ability to analyze, inspect, explore and draw conclusions from the models, all in one integrated framework.

2 Design and implementation

Mackinac provides support for using the ModelSEED web service to create draft GEMs from public genomes available in, or uploaded by the user to, the Pathosystems Resource Integration Center (PATRIC) (Wattam, 2014) and creates a COBRA model from a ModelSEED model. Before using the ModelSEED web service, the user must be a registered PATRIC user and obtain an authentication token using their PATRIC username and password. The get_token() function retrieves and stores the authentication token in the .patric_ config file in the user’s home directory. The user can use this token until it expires.

There are three main functions to reconstruct a GEM using ModelSEED and prepare them for analysis in COBRApy:

- reconstruct_modelseed_model(genome_id, model_id=None)
Description: Reconstructs a draft GEM for an organism. This function requires a PATRIC genome ID to identify the organism; the user can search for genomes on the PATRIC website from the thousands of bacterial organisms available. After a model is reconstructed, it is referred to by an ID. By default, the ID of the model is the PATRIC genome ID.
- gapfill_modelseed_model(model_id, media_reference=None)
Description: Gap fills the draft GEM using the ModelSEED algorithm; requires the model ID. By default, the model is gap filled on a complete medium. Use the media_reference parameter to specify a different growth medium.

- create_cobra_model_from_modelseed_model(model_id)
Description: Creates a COBRA model from the ModelSEED model; requires the model ID. The ModelSEED model is converted to a COBRA model that can be analyzed using all of the functionality in COBRApy.

Additional functions are available for working with ModelSEED models, managing workspace objects, getting information about PATRIC genomes (Table 1). All the functions available are listed in the Mackinac documentation, available in the project folder (<https://github.com/mmundy42/mackinac>).

While outside the scope of functionalities present in Mackinac, it should be noted that it is the researchers’ responsibility to validate models reconstructed automatically and the users should run checks on accuracy of the gapfilling performed using either the ModelSEED or COBRApy framework, the removal of thermodynamic infeasibility loops and the adequacy of the biomass equation used.

3 Conclusion

The rapid reconstruction of GEMs using ModelSEED and the powerful analysis features in COBRApy enable the comprehensive study and exploration of the metabolic function of organisms. Now, Mackinac makes it easy to use the ModelSEED web service to create GEMs that can be seamlessly analyzed with COBRApy. This significantly streamlines the workflow required to explore the large number of species that make up microbial communities.

Acknowledgements

The authors thank all members of the Chia Lab for discussions, critical reading of the manuscript and testing of the package. They also thank Kristin Harper for editorial assistance.

Funding

This work has been supported by the Mayo Clinic Center for Individualized Medicine and the National Institutes of Health [R01CA179243 to N.C.].

Conflict of Interest: none declared.

References

- Benedict, M.N. *et al.* (2014) Likelihood-based gene annotations for gap filling and quality assessment in genome-scale metabolic models. *PLoS Comput. Biol.*, **10**, e1003882.
- Chaouiya, C. *et al.* (2013) SBML Qualitative Models: a model representation format and infrastructure to foster interactions between qualitative modeling formalisms and tools. *BMC Syst. Biol.*, **7**, 135.
- Ebrahim, A. *et al.* (2013) COBRApy: CONstraints-Based Reconstruction and Analysis for Python. *BMC Syst. Biol.*, **7**, 5.
- Henry, C.S. *et al.* (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nat. Biotechnol.*, **28**, 977–982.
- Kumar, V.S. and Maranas, C. (2009) GrowMatch: An automated method for reconciling *in silico/in vivo* growth predictions. *PLoS Comput. Biol.*, **5**, e1000308.
- Reed, J.L. *et al.* (2006) Systems approach to refining genome annotation. *Proc. Natl. Acad. Sci. U. S. A.*, **103**, 17480–17484.
- Thiele, I. and Palsson, B.O. (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat. Protoc.*, **5**, 93–121.
- Thiele, I. *et al.* (2014) FastGapFill: efficient gap filling in metabolic networks. *Bioinformatics*, **30**, 2529–2531.
- Wattam, A.R. *et al.* (2014) PATRIC, the bacterial bioinformatics database and analysis resource. *Nucleic Acids Res.*, **42**, D581–D591.