bioRxiv preprint doi: https://doi.org/10.1101/151258; this version posted June 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

ProbAnnoWeb and ProbAnnoPy: probabilistic annotation and gap-filling of metabolic reconstructions

Brendan King[1], Terry Farrah[1], Matthew Richards[1], Michael Mundy[2], Evangelos Simeonidis[1,*], and Nathan D. Price[1,*]

[1]Institute for Systems Biology, 401 Terry Avenue North, Seattle, WA, 98102, USA, [2]Center for Individualized Medicine, Mayo Clinic, 200 First St. SW, Rochester 55905, MN, USA.

*To whom correspondence should be addressed.

Abstract

Summary: Gap-filling is a necessary step to produce quality genome-scale metabolic reconstructions capable of flux-balance simulation. Most available gap-filling tools use an organism-agnostic approach, where reactions are selected from a database to fill gaps without consideration of the target organism. Conversely, our likelihood based gap-filling with probabilistic annotations selects candidate reactions based on a likelihood score derived specifically from the target organism's genome. Here, we present two new implementations of probabilistic annotation and likelihood based gap-filling: a web service called ProbAnnoWeb, and a standalone python package called ProbAnnoPy.

Availability and Implementation: Our tools are available as a web service with no installation needed (ProbAnnoWeb), available at http://probannoweb.systemsbiology.net, and as a local python package implementation (ProbAnnoPy), available for download at http://github.com/PriceLab/probannopy.

Contact: Evangelos.Simeonidis@systemsbiology.org; Nathan.Price@systemsbiology.org

1 Introduction

Metabolic modeling approaches provide powerful analytical tools for exploration and detailed consideration of the structure and design of a metabolic network (Schilling et al. 1999). Genome-scale models (GEMs) in particular, which collect all available metabolic knowledge on a particular organism, have been constructed for an expanding array of organisms based on annotated genome sequences (King et al. 2016). GEMs have applications in metabolic engineering, modeling of microbial communities, and simulations that combine transcriptomics, proteomics, and/or metabolomics to deepen understanding of an organism's phenotype (Milne et al. 2009; Oberhardt et al. 2009).

When a model is not readily available for an organism, or when existing models are not detailed enough to cover the required elements of metabolism for the intended analysis, a new reconstruction needs to be built. Metabolic reconstruction is a data intensive but well defined process (Thiele and Palsson 2010) that requires collecting species-specific information from genome annotations, high-throughput experiments, the literature, and publically available databases, such as KEGG (Kanehisa et al. 2008) or EcoCyc (Karp et al. 2005). Gap-filling methods (Reed et al. 2006) are subsequently applied to improve connectivity to the point where the model can simulate steady state reaction flux and growth.

Most gap-filling tools use an organism-agnostic approach; one that does not consider the relationship between genome and metabolism in selecting candidate reactions from a database. One such example is parsimonious gap-filling, which fills the model using a universal database such as ModelSEED with as few reactions as possible (Devoid et al. 2013). Conversely, our likelihood based gap-filling (Benedict et al. 2014) uses probabilistic annotation to compute organism-specific reaction likelihoods of gene functions based on sequence homology with a trusted annotation database. These likelihoods can subsequently be used to select gap-filling reactions from a biochemical reaction database (Fig. 1). These annotations can additionally provide insight into an organism's metabolic capabilities and be used in other down-stream modeling tasks.

Here, we provide two new implementations of our annotation likelihood algorithm and its application to likelihood based gap-filling: Pro-

bioRxiv preprint doi: https://doi.org/10.1101/151258; this version posted June 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

bAnnoWeb, a web service, and ProbAnnoPy, a downloadable python package. Our tools are compatible with openCOBRA packages for constraint-based reconstruction and analysis (Schellenberger et al. 2011). Mackinac, a recent tool bridging COBRApy and ModelSEED, additionally provides functions for generating reaction likelihoods and gap-filling models within ModelSEED, as well as transferring ModelSEED models to and from COBRApy (Mundy et al. 2017). We propose our tools as an accessible, easy to use, standalone version of probabilistic annotation, with direct integration with COBRApy for likelihood based gap-filling, and as a minimalist alternative for those who wish to compute locally or to have less interaction with online modeling services.

2 Methods

2.1 Probabilistic Annotation and Reaction Likelihoods

Given an organism's genome sequence, probabilistic annotation assigns an organism-specific likelihood score ($0 \leq s \leq 1$) to each reaction in a template model database of reactions, which comprises the complete pool of candidate reactions for the gap-filling problem. Below we provide a quick overview of this process, which can be explored in greater detail in (Benedict et al. 2014).

First, we run BLASTp on each gene in the query genome against a reference set of high confidence functional annotations (Altschul et al. 1990; Camacho et al. 2009). A log score for each query/target gene pair is computed as follows:

$$S_{ij} = -\log(E_{ij} + k) \tag{1}$$

where E_{ij} is the BLASTp E-value of the pair and k is a small constant (10^{-200}).The probability that a query gene i is in the set of genes A_a with functional annotation a is proportional to the score between query i and each reference target j in A_a:

$$p(i \in A_a) = \frac{\frac{\sum_{j \in A_a} s_{ij}^2}{M}}{\frac{\sum_j s_{ij}^2}{M} + PC} \tag{2}$$

where M is the maximum log-score of any BLASTp hits and PC is a pseudo-count that dilutes likelihoods for annotations with weak homology to the query.

A reaction's likelihood is a function of its corresponding annotation likelihoods derived from its Gene-Protein-Reaction relationship specified in a ModelSEED (Overbeek et al. 2005) template model. For isoenzymes (i.e. "OR" relationships) we take the maximum of enzyme likelihoods, whereas for multi-enzyme complexes (i.e. "AND" relationships) we take the minimum.

2.2 Probabilistic Annotation and Gap-filling

Gap-filling can be formulated as a mixed integer linear programming (MILP) problem: the reactions in the model are considered in union with those in a universal template model, a non-zero or minimum increase constraint is placed on the model's objective function, and the count of new reactions carrying non-zero flux is minimized. In parsimonious gap-filling, each reaction (x) not found in the model receives a gap-filling objective coefficient of one ($\lambda_{gapfill,x} = 1$), and each reaction in the model receives a coefficient of zero.

Likelihood based gap-filling re-weighs the objective coefficients for database reactions according to likelihood as follows:

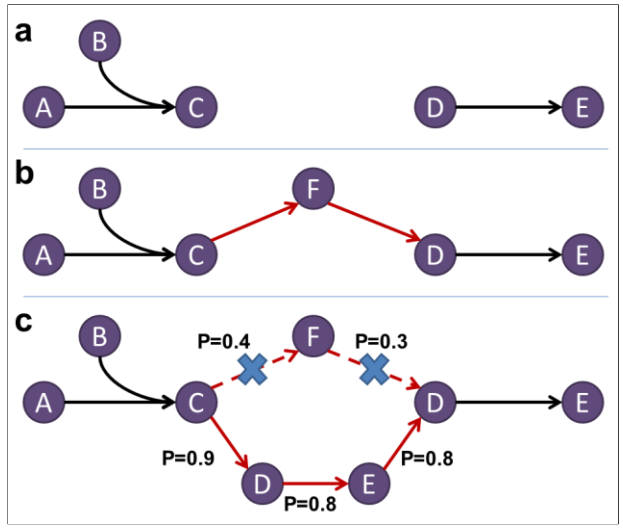

Fig. 1. Likelihood based gap-filling: a. Gap in the metabolic network, preventing production of E; b. A parsimonious gap-filling solution using two new reactions; c. A likelihood based gap-filling solution using three new reactions, which are together more likely than the parsimonious solution.

$$\lambda_{gapfill,x} = \max(1 - p(x), 0) \tag{3}$$

As such, solutions composed of higher likelihood reactions become more favorable, despite potentially requiring more reactions than the optimal parsimonious solution.

3 Workflow

We make two tools available: ProbAnnoWeb, a web service, and ProbAnnoPy, an installable python package. The latter is compatible with Python 2.7, models in COBRApy (Ebrahim et al. 2013) format, and depends on usearch, which is freely available for academic use (Edgar 2010). Like COBRApy, ProbAnnoPy depends on an installed MILP solver such as Gurobi. Greater detail on dependencies and the installation process is available at our GitHub repository.

3.1 Generating Reaction Likelihoods

In the general use case, the workflow begins with finding a genome sequence for a target organism by downloading its proteome sequence in FASTA format. Next, we run probabilistic annotation, which takes the FASTA sequence and a template model as arguments. The template models serve as general databases for reactions and come from ModelSEED. We supply template models for "Gram Positive", "Gram Negative", and "Microbial" organisms. Template choice does not affect likelihoods, only the reactions for which scores are calculated. Probabilistic annotation returns likelihoods in the form of a 'ReactionProbabilities' object, which is a wrapper for a dictionary of reaction likelihoods and other information, such as complex and annotation likelihoods.

3.2 Likelihood based Gap-filling

We provide functionality for likelihood based gap-filling given a model in COBRApy format, a choice of "universal" model to serve as reaction database, and reaction likelihoods. We support functionality for building a "universal" model from one of the supplied template models,

bioRxiv preprint doi: https://doi.org/10.1101/151258; this version posted June 16, 2017. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

a step that is automated behind the scenes in the web service. Although COBRA is identifier-agnostic, our implementations of probabilistic annotation use ModelSEED identifiers; currently, only these identifiers are directly supported for probabilistic annotation and likelihood based gap-filling. Like COBRApy's native parsimonious gap-filling, we output a list of reactions that can be added to the model. ProbAnnoWeb additionally automates this step, instead outputting a gap-filled model that can be downloaded in SBML format.

4 Discussion

Probabilistic annotation is a useful tool both for the analysis of metabolic networks and for likelihood based gap-filling, resulting in higher quality reconstructions corresponding to more genomic evidence. Here, we make freely available to the community a straightforward implementation of this algorithm, which can be used to gap-fill metabolic models with ModelSEED reactions. We provide multiple interfaces to our implementation for varied technical needs and levels of programming savvy. Further work will extend applications of probabilistic annotation and support alternative identifier paradigms.

Acknowledgements

The authors thank Nicholas Chia for important discussions and support.

Funding

This work was supported by the United States Department of Energy's Advanced Research Projects Agency-Energy [grant number DE-AR0000426 to N.D.P.] and the Mayo Clinic Center for Individualized Medicine [M.M.].

Conflict of Interest: none declared.

References

Altschul, S. F., et al. (1990), 'Basic local alignment search tool', *J Mol Biol*, 215 (3), 403-10.

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 (10), e1003882.

Camacho, C., et al. (2009), 'BLAST+: architecture and applications', *BMC Bioinform*, 10, 421.

Devoid, S., et al. (2013), 'Automated genome annotation and metabolic model reconstruction in the SEED and Model SEED', *Methods Mol Biol*, 985, 17-45.

Ebrahim, A., et al. (2013), 'COBRApy: COnstraints-Based Reconstruction and Analysis for Python', *BMC Syst Biol*, 7, 74.

Edgar, R. C. (2010), 'Search and clustering orders of magnitude faster than BLAST', *Bioinformatics*, 26 (19), 2460-61.

Kanehisa, M., et al. (2008), 'KEGG for linking genomes to life and the environment', *Nucleic Acids Res*, 36 (Suppl. 1), D480-D84.

Karp, P. D., et al. (2005), 'Expansion of the BioCyc collection of pathway/genome databases to 160 genomes', *Nucleic Acids Res*, 33 (19), 6083-89.

King, Z. A., et al. (2016), 'BiGG Models: A platform for integrating, standardizing and sharing genome-scale models', *Nucleic Acids Res*, 44 (D1), D515-D22.

Milne, C. B., et al. (2009), 'Accomplishments in genome-scale in silico modeling for industrial and medical biotechnology', *Biotechnol J*, 4 (12), 1653-70.

Mundy, M., Mendes-Soares, H., and Chia, N. (2017), 'Mackinac: a bridge between ModelSEED and COBRApy to generate and analyze genome-scale metabolic models', *Bioinformatics*, doi: 10.1093/bioinformatics/btx185.

Oberhardt, M. A., Palsson, B. O., and Papin, J. A. (2009), 'Applications of genome-scale metabolic reconstructions', *Mol Syst Biol*, 5, 320.

Overbeek, R., et al. (2005), 'The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes', *Nucleic Acids Res*, 33 (17), 5691-702.

Reed, J. L., et al. (2006), 'Systems approach to refining genome annotation', *Proc Natl Acad Sci U S A*, 103 (46), 17480-84.

Schellenberger, J., et al. (2011), 'Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0', *Nat Protoc*, 6 (9), 1290-307.

Schilling, C. H., et al. (1999), 'Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era', *Biotechnol Prog*, 15 (3), 296-303.

Thiele, I. and Palsson, B. O. (2010), 'A protocol for generating a high-quality genome-scale metabolic reconstruction', *Nat Protoc*, 5 (1), 93-121.