Machine Learning Methods for Metabolic Pathway Prediction

Joseph M. Dale, Liviu Popescu, and Peter D. Karp

Pathway Tools Workshop
August 27, 2009

Outline

1. PathoLogic
2. Machine Learning Methods for Prediction
3. Evaluation
4. Conclusions and Future Directions

Pathway Tools Inference Capabilities

- Initial construction, update:
  - Enzyme/reaction matching
  - Pathway prediction
- Refinement:
  - Transcription unit (operon) prediction
  - Transport inference
  - Pathway hole filling

PathoLogic PGDB Construction

- Enzyme names, EC numbers and GO terms from genome annotation are used to identify matching reactions in MetaCyc.
- All MetaCyc pathways with at least one reaction present in the target organism are imported as candidate pathways.
- Candidate pathways are pruned using an iterative algorithm.

PathoLogic Pathway Prediction

PathoLogic uses an iterative algorithm to prune the initial set of candidate pathways:

1. Initialize pathway sets *keep* $= \{\}$, *delete* $= \{\}$, *undecided* $=$ all initial candidates.
2. Apply "keep tests" $K_1, \ldots, K_m$ to *undecided* pathways; if any $K_i(p)$ succeeds, move p to *keep* set.
3. Apply "delete tests" $D_1, \ldots, D_n$ to *undecided* pathways; if any $D_i(p)$ succeeds, move p to *delete* set.
4. If any *undecided* pathways were moved, update pathway evidence and go to step 2; otherwise terminate.

keep pathways and remaining *undecided* pathways (no keep or delete tests succeeded) are kept in PGDB.

Examples of Keep Tests

- pathway has a unique reaction present
- pathway is “mostly present”:
  - at most one reaction missing
  - more reactions present than missing
  - evidence not a proper subset of evidence for variant
  - not a superset of another pathway
- pathway evidence is not a subset of evidence for any other pathway, and pathway is not missing all key reactions (curated in MetaCyc)

Examples of Delete Tests

- pathway “mostly absent”:
  - at most one reaction present
  - more than one reaction missing
  - no unique reactions present
- biosynthetic pathway missing final steps
- degradative pathway missing initial steps
- pathway missing all “key reactions”

Limitations of PathoLogic

- As MetaCyc grows (currently $>$ 1300 pathways), PathoLogic makes more false positive predictions
- Okay for PGDBs that will receive manual curation (this was intended), but problematic for BioCyc PGDBs that receive no curation
- Several areas in which PathoLogic is limited:
    - *extensibility*
    - *tunability*
    - *explainability*

Extensibility

- Above description of PathoLogic above is a simplification! The actual logic is more complex, hard-coded, and brittle.
- Difficult to add new tests (keep and delete rules), specify interactions with existing tests.
- No formal training procedure to incorporate feedback (i.e., automatically adjust to correct false predictions).

Tunability

- PathoLogic currently only makes binary predictions (pathway present / absent).
- Can't be tuned to trade off sensitivity/specificity, precision/recall – performance is fixed at a single point.
- Preference for false positives is hard-coded.

Explainability

- Existing confidence scores are coarse: e.g., fraction of reactions present, number of unique enzymes.
- Not monotonic: pathway X may have more reactions present than pathway Y, but X can be pruned while Y is kept.
- Users can’t see how evidence was combined: which rules were applied to call the pathway present / absent.

The Machine Learning Approach

Supervised machine learning:

- Collect training data:
  input feature (attribute) vectors $X_1, \ldots, X_n$
  output labels $y_1, \ldots, y_n$
- Apply learning algorithm to training data, obtain structure, parameters of function $F : X \rightarrow y$.
- Apply F to new feature vector X_{n+1} to yield prediction $\hat{y}_{n+1} = F(X_{n+1})$

Machine Learning Approach to Pathway Prediction

- Collect a “gold standard” set of labeled data for training (and validation): known data on pathway presence/absence in various organisms.
- Define useful features; compute feature values for each pathway.
- Input the feature data to domain-independent learning algorithm to train a model for pathway prediction.
- Apply the model to new pathway examples when building a new PGDB.

Can Machine Learning Help?

- ML methods have automated training procedures, easy to add new features and training data.
- Many ML methods have probabilistic foundations, yielding natural confidence scores:
  $Pr(pathway\ present \mid evidence)$.
- Many ML methods can explain predictions; e.g., log-likelihood score for each feature, etc.

Feature Extraction

Features are the primary domain-specific component of ML models. Ours fall into several groups:

- **Reaction evidence**: based on matching pathway reactions to enzymes based on genome annotation; e.g., fraction of reactions present; number of unique enzymes.
- **Pathway holes**: patterns of pathway holes (reactions missing enzymes); e.g., biosynthetic pathway missing final reactions; degradation pathway missing initial reactions.
- **Genome context**: e.g., two reactions in pathway encoded by genes adjacent on chromosome?

Feature Extraction

More feature groups:

- **Pathway variants**: e.g., is the evidence for pathway V_1 a subset of the evidence for its variant V_2?
- **Taxonomic range**: does the expected taxonomic range of the pathway (curated in MetaCyc) include the target organism?
- **Pathway connectivity**: e.g., number of dead end compounds in the pathway, number of adjacent pathways (*via* input/output metabolites)
- **Miscellaneous PathoLogic features**: other features adapted from PathoLogic.

Feature Selection

- In total, 123 features were defined – many slight variations. Multiple redundant features can degrade the performance of some ML methods.
- Experimented with various feature selection methods: Akaike information criterion (AIC), Bayes information criterion (BIC), cross-validation.
- Simple hill-climbing on AIC performed as well as more sophisticated (and slower) methods.

Prediction Methods

Different ML methods perform better on different problems. We evaluated several methods:

- naïve Bayes
- decision trees
- logistic regression
- *k* nearest neighbors
- ensemble methods:
  - bagging
  - boosting
  - random forests

Gold Standard Dataset

- Training / validation set based on six curated PGDBs: *E. coli*, *Arabidopsis*, yeast, mouse, cattle, *Synechococcus elongatus*
- 5,610 tuples of the form $(organism, pathway, present|absent)$
- Positive (present) examples are those pathways included in PGDB after curator review.
- Negative (absent) examples include pathways deleted by curators and pathways with no reactions present.

Gold Standard Dataset

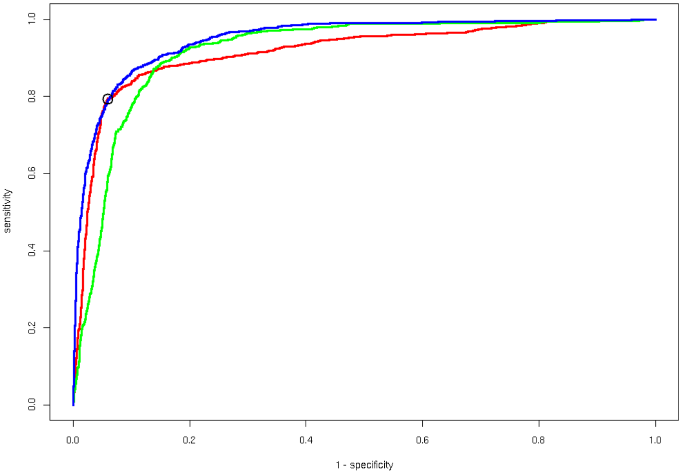
Breakdown of gold standard pathways by organism:

organism	positives	negatives	total
Escherichia coli K-12 MG1655	235	1035	1270
Arabidopsis thaliana columbia	297	971	1268
Saccharomyces cerevisiae S288c	119	777	896
Synechococcus elongatus PCC 7942	171	778	949
Mus musculus	203	754	957
Bos taurus	151	119	270

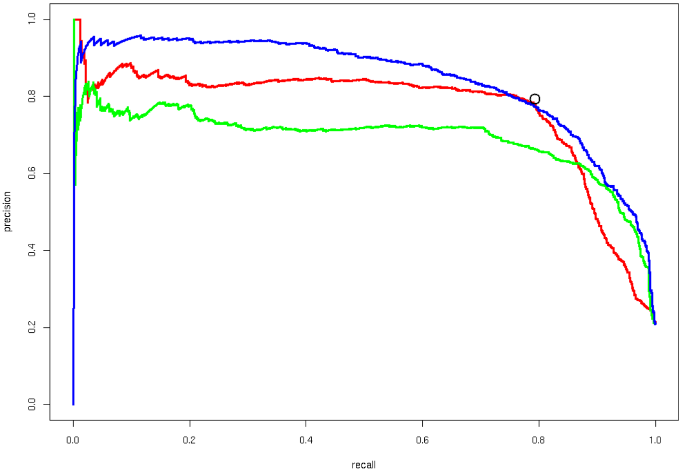
Validation Methodology

- Learning curves: 80%/20% training/test split; select subsets of training set, varying size; measure on test set
- Overall performance measured on random 50%/50% training/test split, repeated and averaged 20x
- cross-validation for ROC curves

PathoLogic vs. ML, ROC on sensitivity/specificity



PathoLogic vs. ML, ROC on precision/recall



PathoLogic vs. ML, optimal threshold

High-performing ML methods vs. PathoLogic:

method	ACC	SN	SP	FM	PR	RC
PathoLogic	0.91	0.793	0.94	0.786	0.779	0.793
naïve Bayes (HC-AIC, 15x bagged)	0.909	0.757	0.949	0.78	0.767	0.796
logistic regression (HC-BIC, 8x bagged)	0.912	0.744	0.956	0.786	0.763	0.812
decision trees (SSMML, 25x bagged)	0.911	0.729	0.961	0.787	0.77	0.808

Conclusions

- Performance of ML algorithms roughly equals that of PathoLogic
- Advantages of ML methods over PathoLogic:
  - numerical confidence scores
  - tradeoff between sensitivity/specificity, precision/recall
  - easily extensible
  - explanation of predictions

Future Work

- Integrate into Pathway Tools
- Improve enzyme name matching
- More sophisticated prediction algorithms, using:
  - dependencies between features
  - iterative refinement
  - dependencies between pathways