Bridging the Phenotype Divide by Using Shared Patterns

Susan M Bello and Nicole A Vasilevsky

University of Colorado
Anschutz Medical Campus

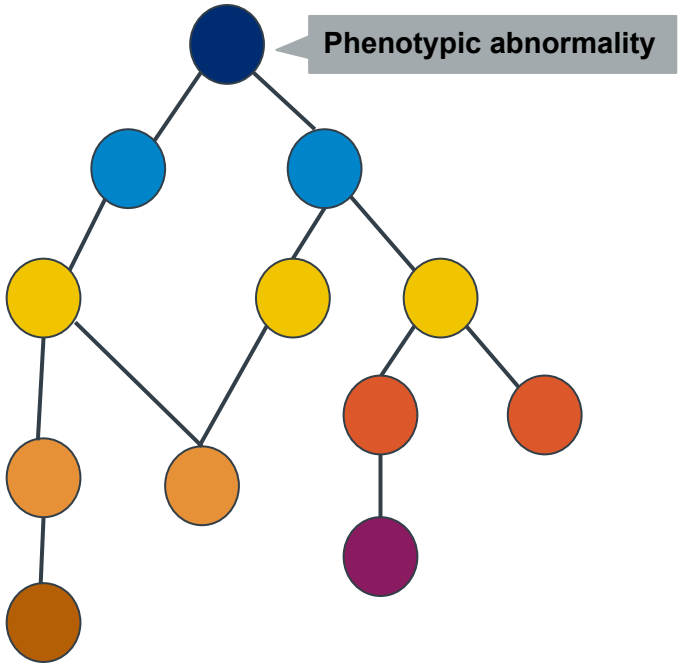
The Jackson Laboratory
Leading the search for tomorrow's cures

The Challenge

Bring together multiple species specific phenotype ontologies while still supporting the individual ontologies

How We Got Here

- Independent development aimed at meeting specific needs of one or a at most a few communities
- Divergent meanings of the word “phenotype”
- Iterative development of logical definitions of terms

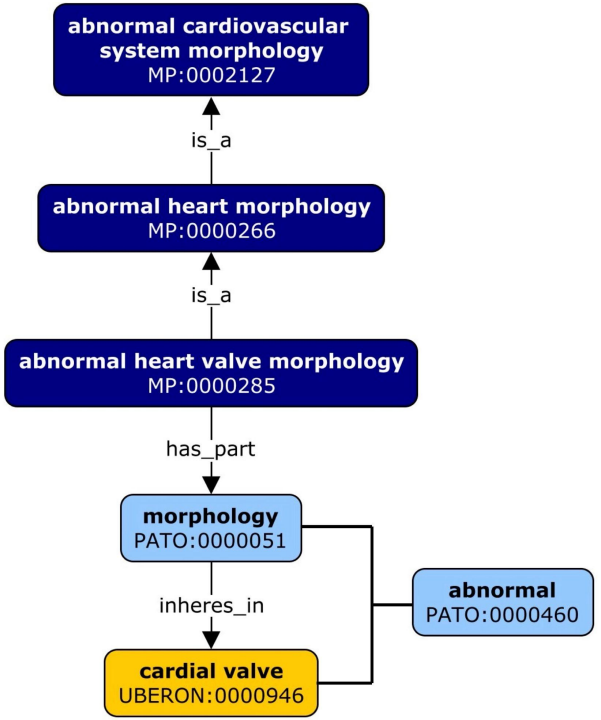


Logical Definitions

- Also referred to as equivalence axioms
- Breaks down a phenotype term into constituent parts
- Pieces of the logical definition come from other ontologies

abnormal heart valve morphology (MP:0000285)

has part some (morphology and inheres in some cardial valve and has modifier some abnormal)



Logical Definitions

- Also referred to as equivalence axioms
- Breaks down a phenotype term into constituent parts
- Pieces of the logical definition come from other ontologies

abnormal heart valve morphology (MP:0000285)

BFO

has part some (morphology and inheres in some cardial valve

and has modifier some abnormal)

Logical Definitions

- Also referred to as equivalence axioms
- Breaks down a phenotype term into constituent parts
- Pieces of the logical definition come from other ontologies

abnormal heart valve morphology (MP:0000285)

PATO

has part some (morphology and inheres in some cardial valve

and has modifier some abnormal))

Logical Definitions

- Also referred to as equivalence axioms
- Breaks down a phenotype term into constituent parts
- Pieces of the logical definition come from other ontologies

abnormal heart valve morphology (MP:0000285)

has part some (morphology and inheres in (RO) some cardial valve

and has modifier some abnormal)

Logical Definitions

- Also referred to as equivalence axioms
- Breaks down a phenotype term into constituent parts
- Pieces of the logical definition come from other ontologies

abnormal heart valve morphology (MP:0000285)

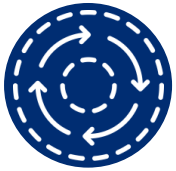
UBERON or species specific anatomy Ontology

has part some (morphology and inheres in some cardial valve

and has modifier some abnormal))

Challenges of Logical Definition Implementation

Maintenance of definitions when changes are made to how a definition is built



Ensuring inter-ontology and inter-curator consistency in creating definitions

Logical Definition Maintenance

A change to the style of building a definition had to be manually made to each term using a similar definition

Absent X | Version 1

```
'has part' some (absent and ('inheres in' some  ([X] and ('part of' some [structure containing X]))) and ('has modifier' some abnormal))
```

Absent X | Version 2

```
'has part' some (absent and ('inheres in' some [X]) and ('has modifier' some abnormal))
```

Inconsistent Building of Definitions

Selection of

- Ontology to use for part of a definition
- Specific term to use within a definition

Anatomical Cyst

Style 1
uses cystic from MPATH

```
'has part' some (cystic and
('inheres in' some [anatomical
entity])
and ('has modifier' some abnormal))
```

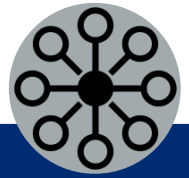
Style 2
uses 'increased amount' and 'cyst' from PATO

```
'has part' some ('increased amount'
and ('inheres in' some (cyst and
('part of' some [anatomical
entity]))) and ('has modifier' some
abnormal))
```

The Solution: How Shared Patterns Meet These Challenges

- What is a Shared Pattern
- How does this address
  - Consistency
  - Maintenance

What is a Shared Pattern



Dead Simple Ontology Design Patterns (DOSDPs)
PMID: 28583177

Developed and agreed to by participating uPheno members

Patterns are freely available on the uPheno GitHub

https://github.com/obophenotype/upheno

Components of a Design Pattern

- Defined set of
  - Ontology classes
  - Relations
  - Variable classes
- Generic version of the pattern using the defined sets

```
'"has_part' some ('cystic' and
('inheres_in' some %s) and ('has_modifier'
some 'abnormal'))"
```

- Additional lines to allow for automatic generation of term definition, name, and synonyms

Example pattern: **Cyst in location**

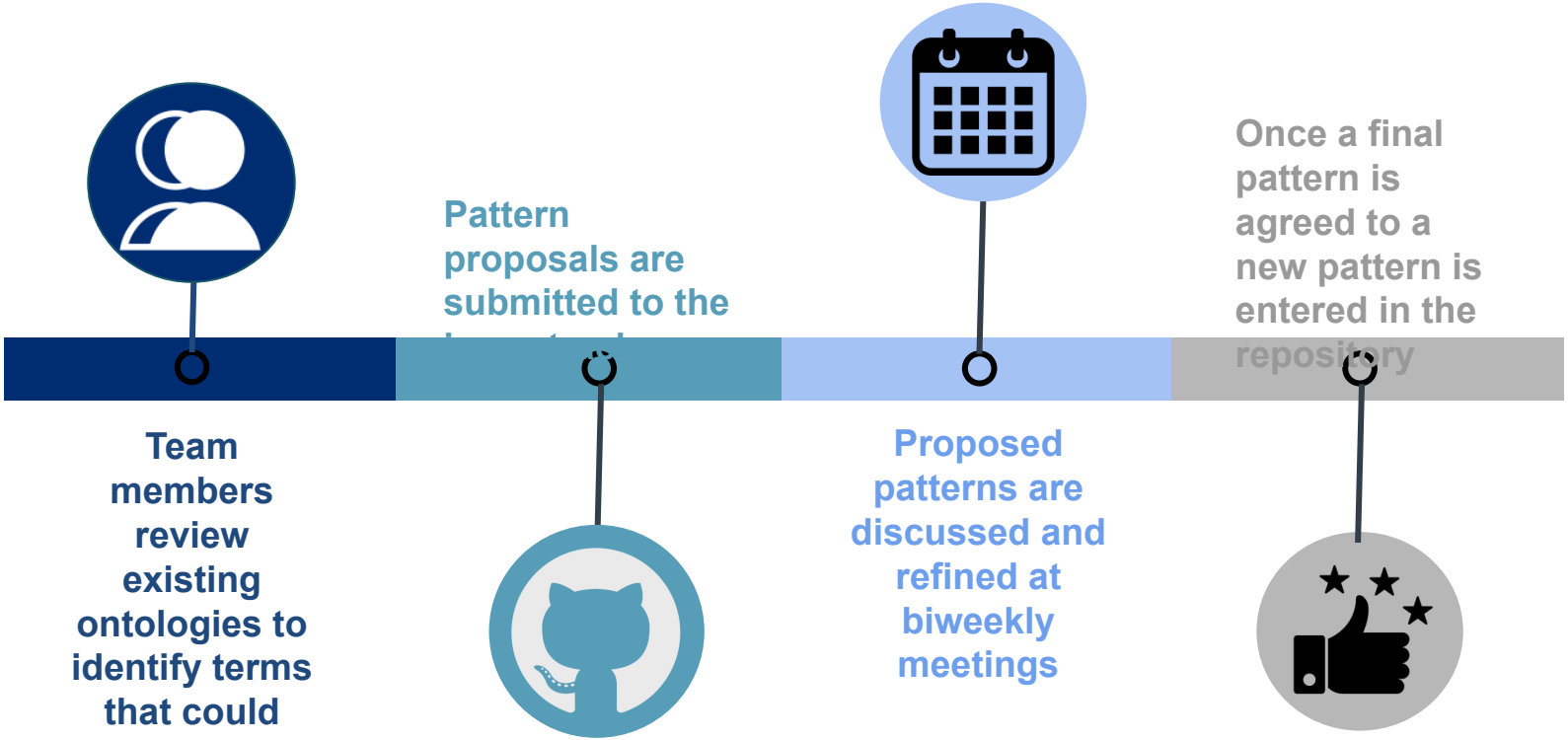
```
pattern_name: cystInLocation
pattern_iri: http://purl.obolibrary.org/obo/upheno/patterns-dev/cystInLocation.yaml
description: "The presence of a cyst in an anatomical entity. For example, HP_0010604 'Cyst of the eyelid'."

contributors:
- https://orcid.org/0000-0001-5208-3432
- https://orcid.org/0000-0002-3528-5267
classes:
  cystic: PATO:0001673
  abnormal: PATO:0000460
  anatomical entity: UBERON:0001062

relations:
  inheres_in: RO:0000052
  has_modifier: RO:0002573
  has_part: BFO:0000051

vars:
  anatomical_entity: "'anatomical entity'"

name:
  text: "%s cyst"
  vars:
  - anatomical_entity

def:
  text: "The presence of a cyst in the %s."
  vars:
  - anatomical_entity

equivalentTo:
  text: "'has_part' some ('cystic' and ('inheres_in' some %s) and ('has_modifier' some 'abnormal'))"
  vars:
  - anatomical_entity
```

HP:0410278 'Pituitary gland cyst'

The presence of a cyst in the pituitary gland.

'has part' some (cystic and ('inheres in' some pituitary gland) and ('has modifier' some abnormal

How Are Patterns Created



https://github.com/obophenotype/upheno

How Are Patterns Used

Incorporated into ontology build process

Build process creates logical definitions for a specified set of terms using the pattern files

New terms can be created using patterns by specifying the pattern and the desired entity to use for the variable

A bridging ontology (uPheno) can be built by using the patterns and the structure of the ontologies used in the patterns

How Do Patterns Improve Consistency and Maintenance

Consistency

- Reduces inter-ontology and inter-curator variation
- Defines the ontology classes and variables thus reducing opportunities for inconsistency

Maintenance

- A change to a pattern is automatically propagated to all terms using the pattern during the build process

Additional benefits

- During pattern development issues with the underlying ontologies (i.e. PATO, Uberon) may be identified and resolved
- Once implemented inferences made using the shared patterns are reviewed to identify any potential problems
  - If a problem is identified the pattern or underlying ontology can be altered as needed to make sure the inferences are correct

How to Find Patterns

Patterns reside in the uPheno GitHub site and can be browsed in the Pattern directory

>200

patterns have been created

uPheno Participants

Ontologies and Databases

- Human Phenotype Ontology
- Mammalian Phenotype Ontology (MGI and RGD)
- Zebrafish Information Network
- Xenopus Phenotype Ontology (XenBase)
- WormBase Phenotype
- Drosophila Phenotype Ontology (FlyBase)
- Fission Yeast Phenotype Ontology (PomBase)
- Plant Trait Ontology
- Phenotype and Trait Ontology (PATO)
- SGD
- dictyBase
- Monarch Initiative
- Alliance of Genome Resources
- Phenoscape
- Planteome, Bioversity

uPheno Participants

Nico's development work is supported by NIH / NHGRI Grant #1RM1HG010860-01

Individuals

- Susan M Bello
- Nicole Vasilevsky
- Leigh C. Carmody
- Yvonne M. Bradford
- Stacia Engel
- Chris Grove
- Clare Pilgrim
- Simon Jupp
- Petra Fey
- David Osumi-Sutherland
- Midori A. Harris
- Valerie Wood
- Peter Robinson
- Sebastian Koehler
- Anna V. Anagnostopoulos
- Wasila Dahdul
- Alayne Cuzick
- Sofia Robb
- Erik Segerdell
- Laurel Cooper
- Marie-Angélique Laporte
- Pankaj Jaiswal
- Chris Mungall
- Melissa Haendel
- Nico Matentzoglu

uPheno calls are on the 2nd and 4th Thursday each month.

Slack workspace: phenotype-ontologies.slack.com

Please contact Sue or Nicole for invitations if you would like to join!