Integration of PO with other OBO ontologies

Chris Mungall

LBL

OBO Foundry

- Multiple ontologies designed to interoperate
- Granularities
  - subcellular
  - cellular
  - organismal
  - population and environment
- Perspective
  - anatomical / static
  - processual / dynamic
- Phenotype
  - “normal” / wild type
  - “abnormal” – disease and mutant phenotypes

Why integrate?

- Ontology maintenance
  - reuse the work of others
  - building block approach
- Data discovery
  - Queries across multiple databases

PO - Get to know your neighbors

- Vertical
  - CL (generic cell types)
  - CARO
- Incoming
  - GO (development)
  - TO (plant trait)
- Outgoing
  - GO (system processes)
  - CHEBI (chemical entities)
  - PRO (protein)
  - Taxonomy (NCBI?)

Integration with CL

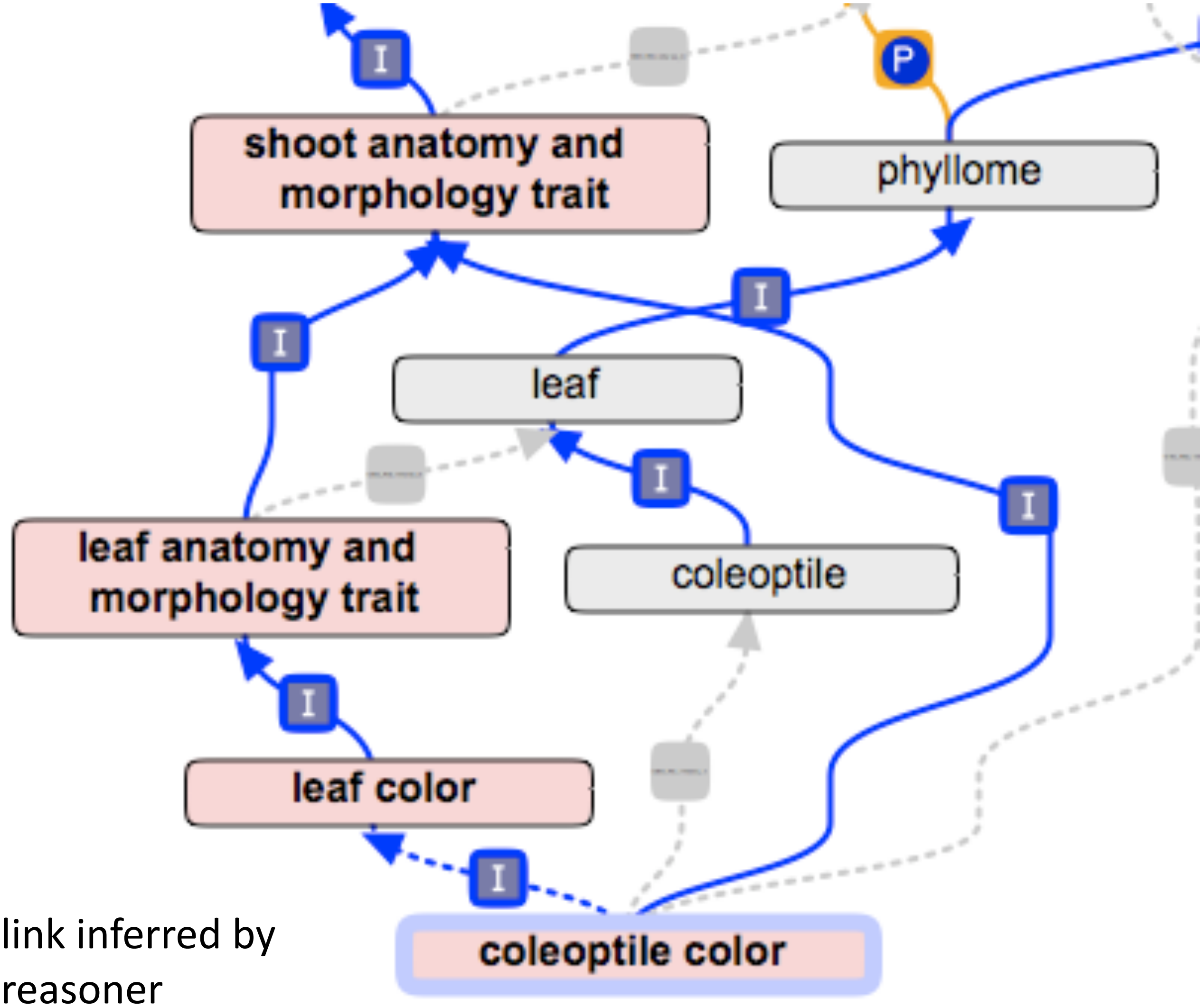
- Some PO classes (like plant cell) are subtypes of generic cell types (like "cell")
- PO responsibility
  - Generate a bridge file to CL
    - Include OBO-unique synonyms
    - can be generated from xrefs
    - Will most likely be rather small!
- CL responsibility
  - Generate the multispecies cell ontology
    - amalgam of generic CL, fly CL, plant CL, …

Integration with plant trait TO (incoming)

- Many phenotypes described anatomically
- Phenotype ontologies can include definitions that use PATO ontology of qualities
  - mammalian_phenotype_xp
  - human_phenotype_xp
  - ascomycetes_phenotype_xp
  - **plant_trait_xp**

Christopher Mungall, Georgios Gkoutos, Cynthia Smith, Melissa Haendel, Suzanna Lewis, and Michael Ashburner**. Integrating phenotype ontologies across multiple species**. *Genome Biology*, 11(1):R2, 2010
http://www.berkeleybop.org/people/cjm/Mungall-GO-JBI-2010.pdf

plant_trait_xp


I
P
shoot anatomy and morphology trait
phyllome
I
I
leaf
I
I
leaf anatomy and morphology trait
coleoptile
I
leaf color
I
coleoptile color

link inferred by reasoner

obtaining plant_trait_xp

- http://www.obofoundry.org/cgi-bin/detail.cgi?id=plant_trait_xp

- maintained in obo sf cvs
  - obo/phenotype/plant_trait/
    - plant_trait_xp.obo
    - plant_trait_xp_mireot.obo

Integration with GO (incoming)

- GO covers development
  - E.g. pollen tune growth
- Many such GO classes are being logically defined using external ontologies
  - biological_process_xp_uber_anatomy
  - **biological_process_xp_plant_anatomy**
  - biological_process_xp_fungal_anatomy

Christopher J. Mungall, Michael Bada, Tanya Z. Berardini, Jennifer Deegan, Amelia Ireland, Midori A. Harris, David P. Hill, and Jane Lomax. **Cross-Product Extensions of the Gene Ontology**. *Journal of Biomedical Informatics* 2010
http://www.berkeleybop.org/people/cjm/Mungall-GO-JBI-2010.pdf

Integration with GO (outgoing)

- GO has biological process classes relevant to plant functioning
  - photosynthesis
  - respiratory gaseous exchange
  - response to wounding
- These could be used to define or to add information about PO classes
  - see also CL

Integration with taxonomies

- PO currently includes 77 sensu terms
- These should be replaced by terms with non-taxonomic differentiae
- taxonomic information should be added as non-defining links
  - only_in_taxon
  - never_in_taxon

- every arabidopsis female gametophyte is a embryo sac

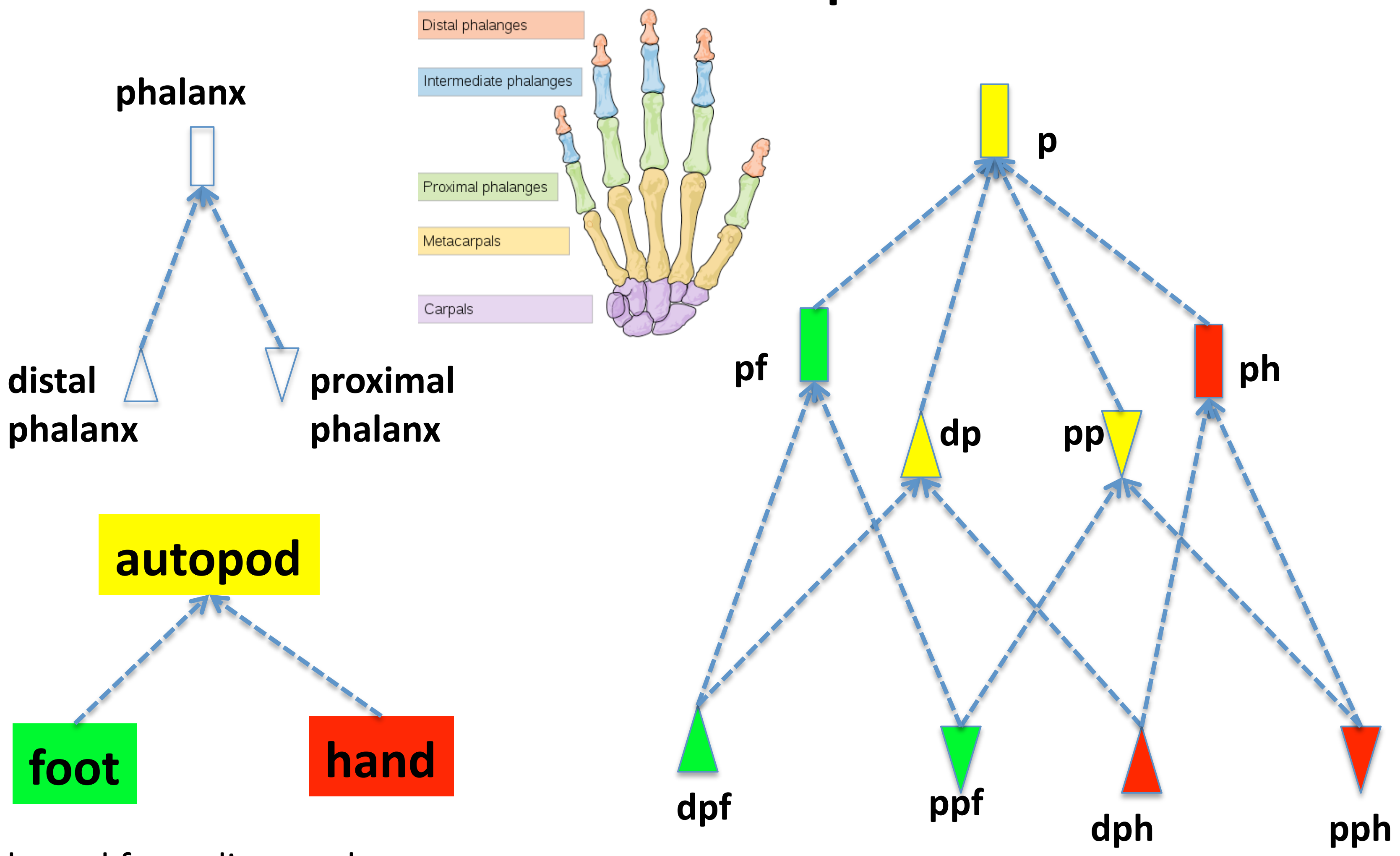
- END

Logical definitions for PO

Why make definitions computable?

- Doing the work all by yourself is hard, boring and error prone
- Automate using reasoning
- Bonus:
  - Makes the definitions easier for humans
  - Enables better visualization

Biology is modular, most ontology classes are compositional



(adapted from diagram by David Hill and Joel Richardson)

Logical definitions

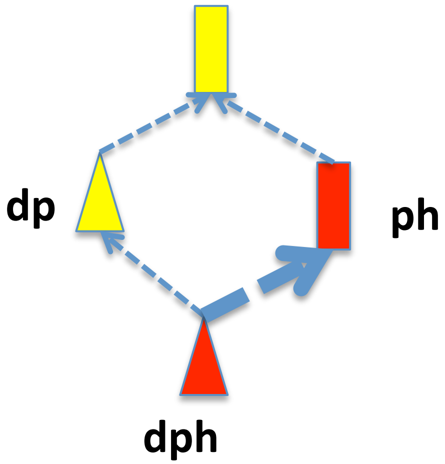
- Genus-differentia form
  - Text definition is genus-differentia form
    - **Distal phalanx of hand:**
      - "A **distal phalanx** that is part of a **hand.**"
      - or "A **distal phalanx**[ID:2] that is part of a **hand**[ID:3]**.**"
  - OWL:
    - **'distal phalanx of hand'** EquivalentTo **'distal phalanx'** and **part_of** some **hand**
  - OBO-Format
    - `[Term]`
    - `id: ID:1 ! `**`distal phalanx of hand`**
    - `intersection_of: ID:2 ! `**`distal phalanx`**
    - `intersection_of: part_of ID:3 ! `**`hand`**

Editing logical definitions

- OBO-Edit
  - Cross-product tab
    - Genus
    - Differentia
  - Parent Editor
    - select links to make them ‘intersections’
- Protégé 4
  - Equivalent Classes
    - Enter expression

Using a reasoner

- Given:
  - **‘distal phalanx of hand’** EquivalentTo **‘distal phalanx’** and **part_of** some **hand**
  - **‘phalanx of hand’** EquivalentTo **phalanx** and **part_of** some **hand**
  - **‘distal phalanx’** is_a **phalanx**
- A reasoner can infer that:
  - **‘distal phalanx of hand’** is_a **‘phalanx of hand’**



Uses of a reasoner

- Ontology authoring
  - – Time saving
    - **Automatically inferring** *is_a* polyhierarchy
  - – Quality Control
    - detecting **inconsistencies**
- Data integration and discovery
  - – Less applications… so far

Reasoners

- OBO-Edit
  - Rule Based Reasoner
- OWL Reasoners (Protégé 4)
  - Pellet
  - FaCT++
  - HermiT

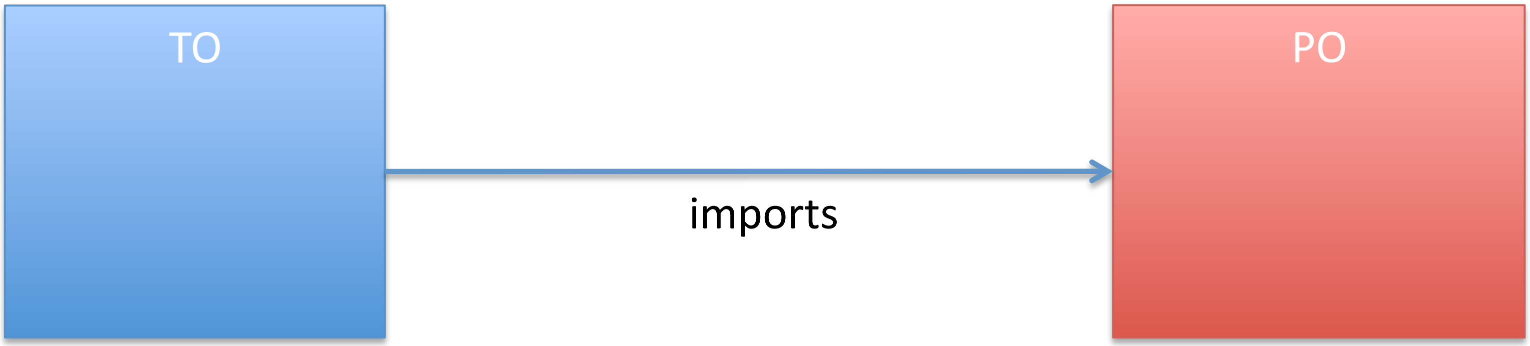
Logical definitions for PO

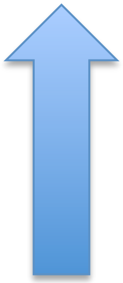
- Most will be internal
  - E.g.
    - tuber cortex = **stem cortex** that is *part of* **some tuber storage parenchyma**
- Some will be external
  - E.g. lactifer PO:0005053
  - References external ontology

Modularity and external ontologies

- Modularity principle:
  - Multiple orthogonal ontologies
  - Use classes from O^1 as building blocks in O^2
  - E.g.
    - *pollen tube* growth
    - anucleate cell
- BUT: Can pose problems for large external ontologies
  - CHEBI
  - PRO

Import

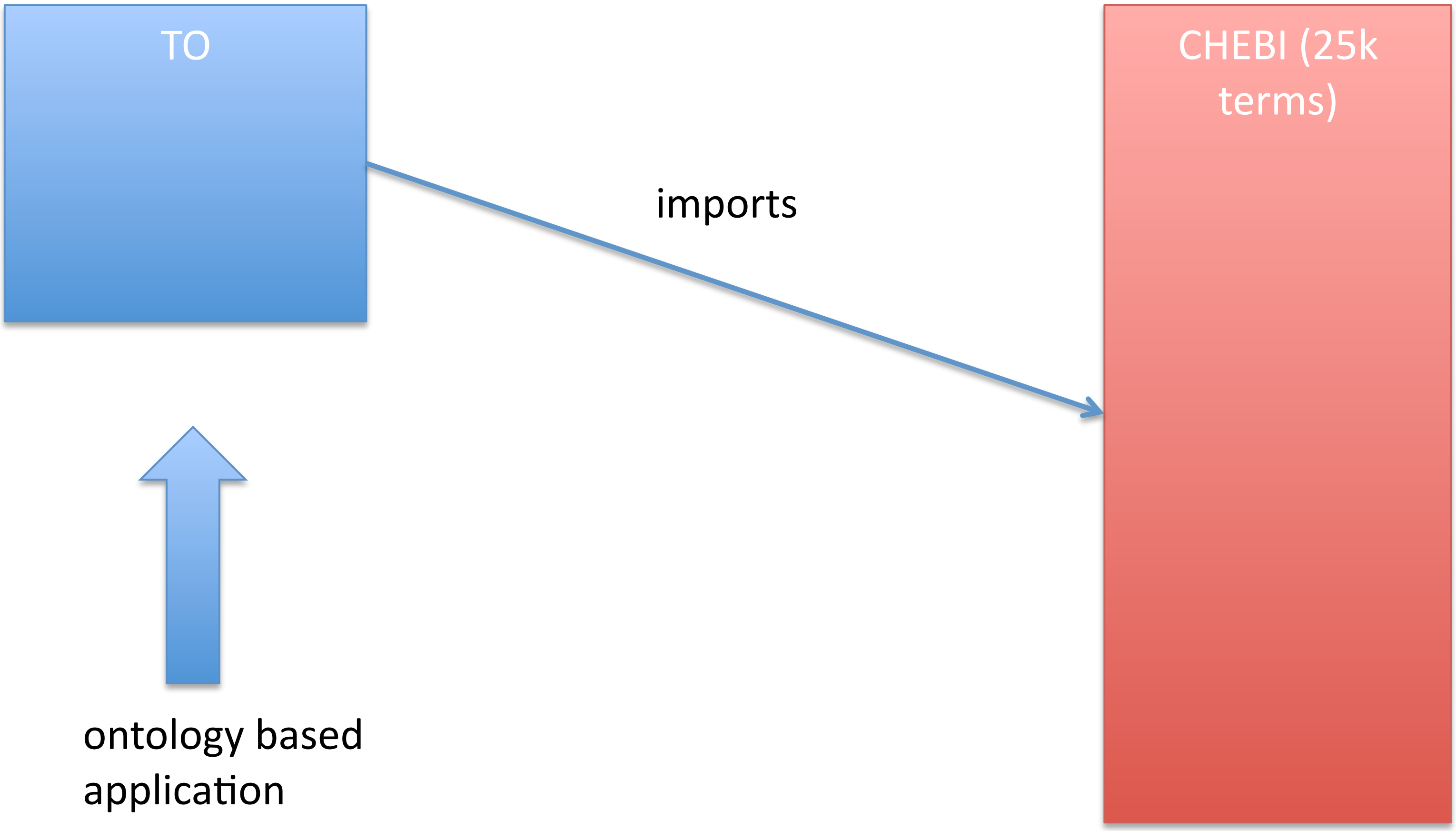




ontology based application

- application can choose to follow import chain
  - if it doesn't, then there are *dangling references*
- the full *import closure* can be large!

Import



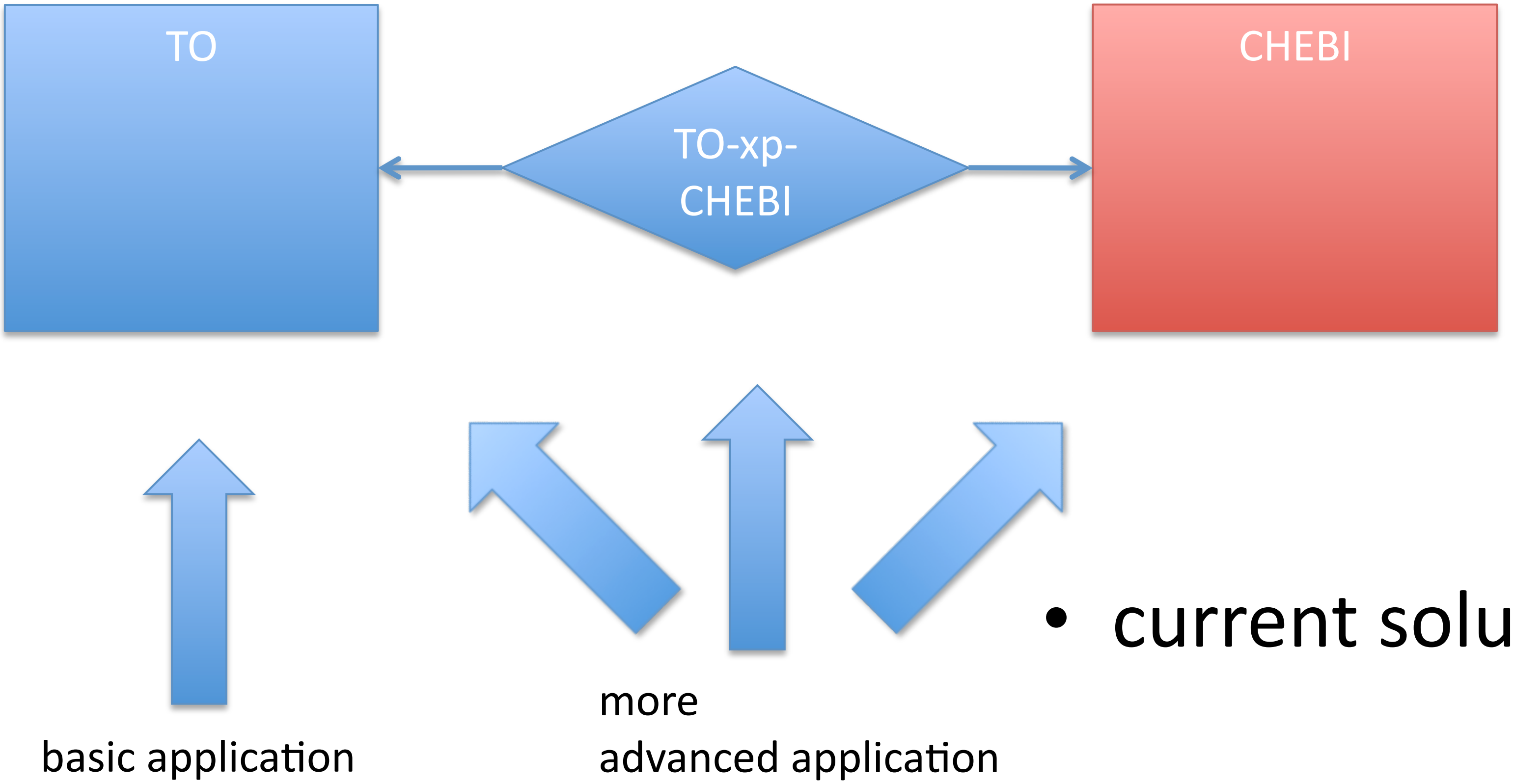
TO

imports

CHEBI (25k terms)

ontology based application

Bridge files



TO

TO-xp-CHEBI

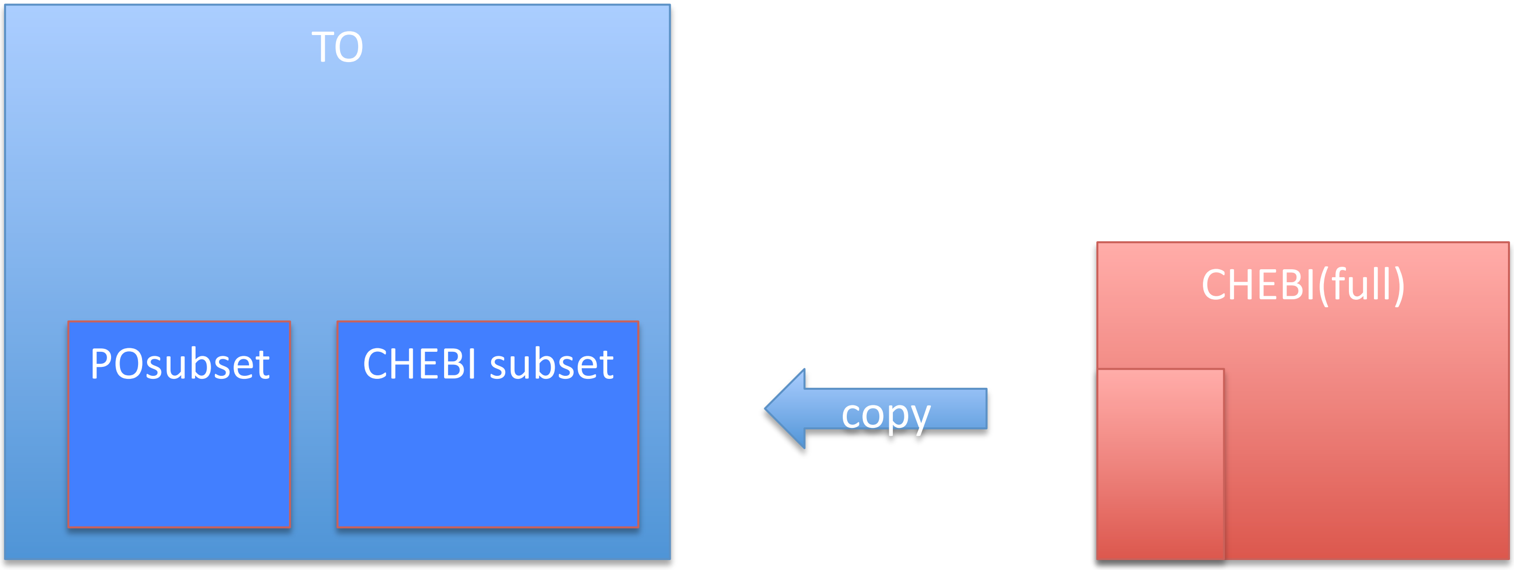
CHEBI

basic application

more advanced application

- current solution

MIREOT - Minimal Information for Retrieval of External Ontology Terms

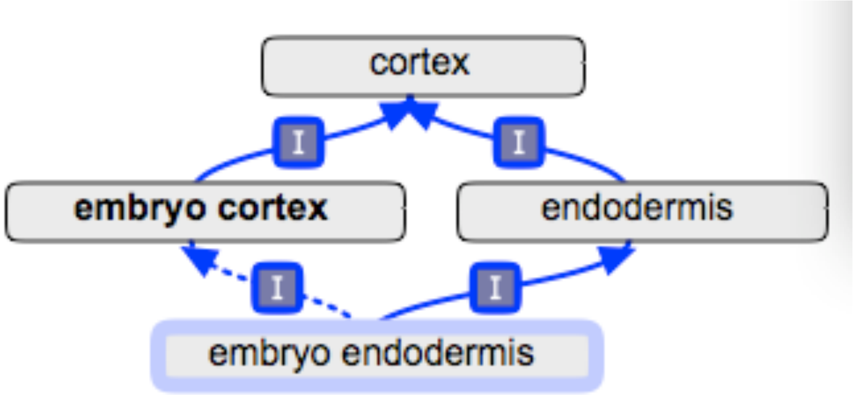


- current solution used by OBI, PRO, hemo-CL

- ontology author "MIREOT"s in subset of external ontology

Deploying po_anatomy

- Separate editors version from deployed version
  - generate deployed version automatically from editors version
  - editors version is single inheritance
  - deployed version is read only
  - deployed version has inferred polyhierarchy



Next steps

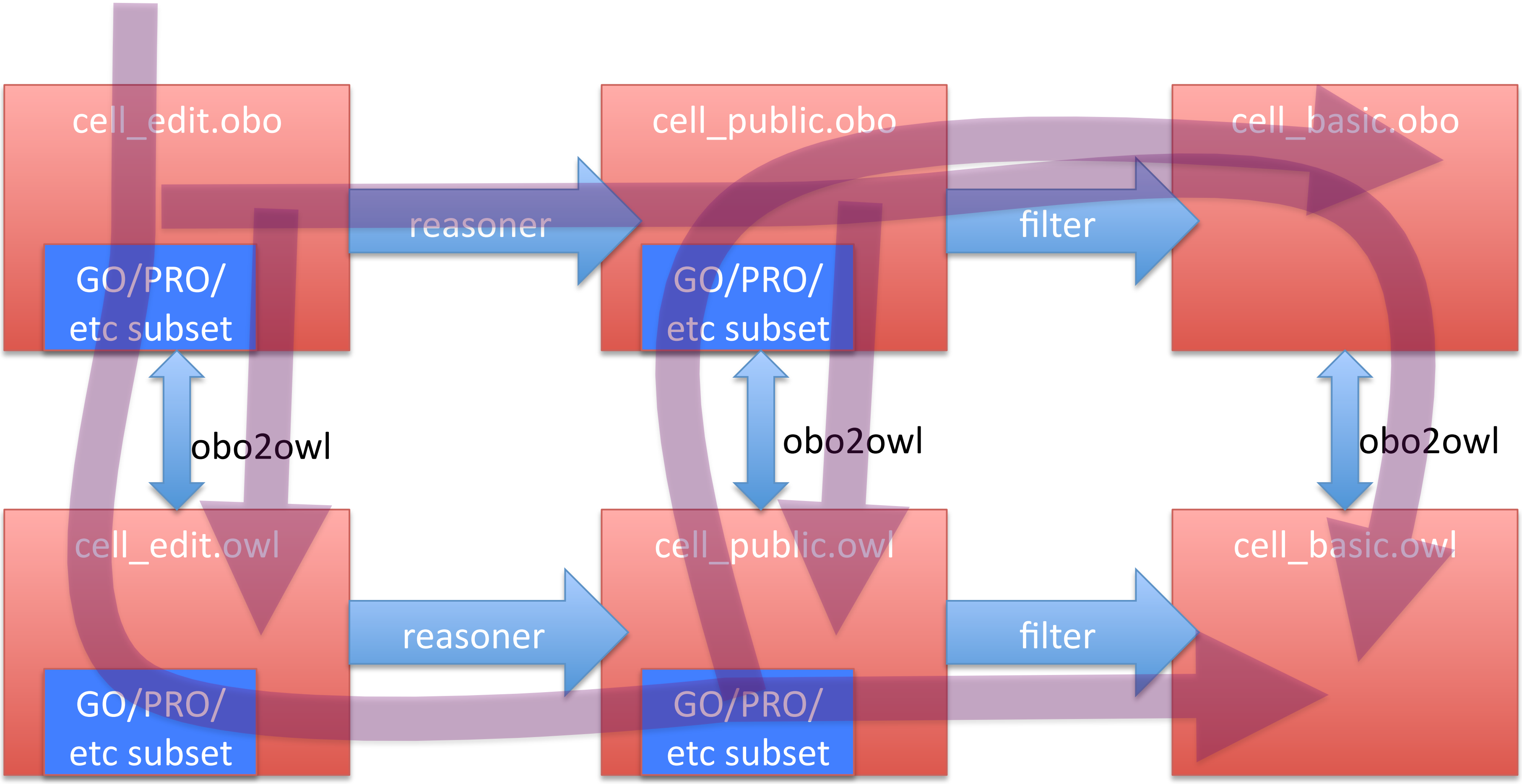
- PO is already specifying logical definitions in the main ontology
  - these all look good
  - these are all part_of differentia so far
- Will we need other ontologies for definitions?
- TO is far more dependent on external ontologies
  - MIREOT approach will be necessary

- END

Deploying cell.obo

- Typical User
  - inferred links materialized
  - external ontology links removed
- Advanced User
  - asserted links only
  - external links present
  - minimal external classes included

Ontology publishing pipeline



Summary

- Many ontologies are seeing the benefits of using computable definitions
  - e.g. fly anatomy
  - GO slow to adopt, lots of legacy issues to work out
- Better to employ them from the outset
  - DC_CL
  - Hemo_CL

- END

Shortcut relations

Relations used

- hemo-CL uses
  - capable_of
  - lacks_part (Ceusters et al)
  - has_plasma_membrane_part (Masci et al)
  - lacks_plasma_membrane_part (Masci et al)
  - has_high_plasma_membrane_amount (Masci et al)
  - has_low_plasma_membrane_amount (Masci et al)

Pros and cons of these relations

- Objections
  - Artificial
    - We should only use a very minimal set of relations from RO, too many relations are bad (why?)
  - Serious
    - Information is hidden from reasoner (example coming later)
- Strengths
  - Simple
  - Intuitive
  - Makes repeated use easier

Short vs long

- What does **has_plasma_membrane_part** mean?
- Shortcut relationship:
  - C **has_plasma_membrane_part** some M
- Better expressed in OWL as:
  - C **has_part** some ('GO:plasma membrane' and **has_part** some M)
    - uses only core relations
    - fully expresses semantics
    - repetitive pattern -> tedious, RSI
    - nested expressions are hard!
      - difficult in OE
      - difficult to load into databases
    - make reasoning slower

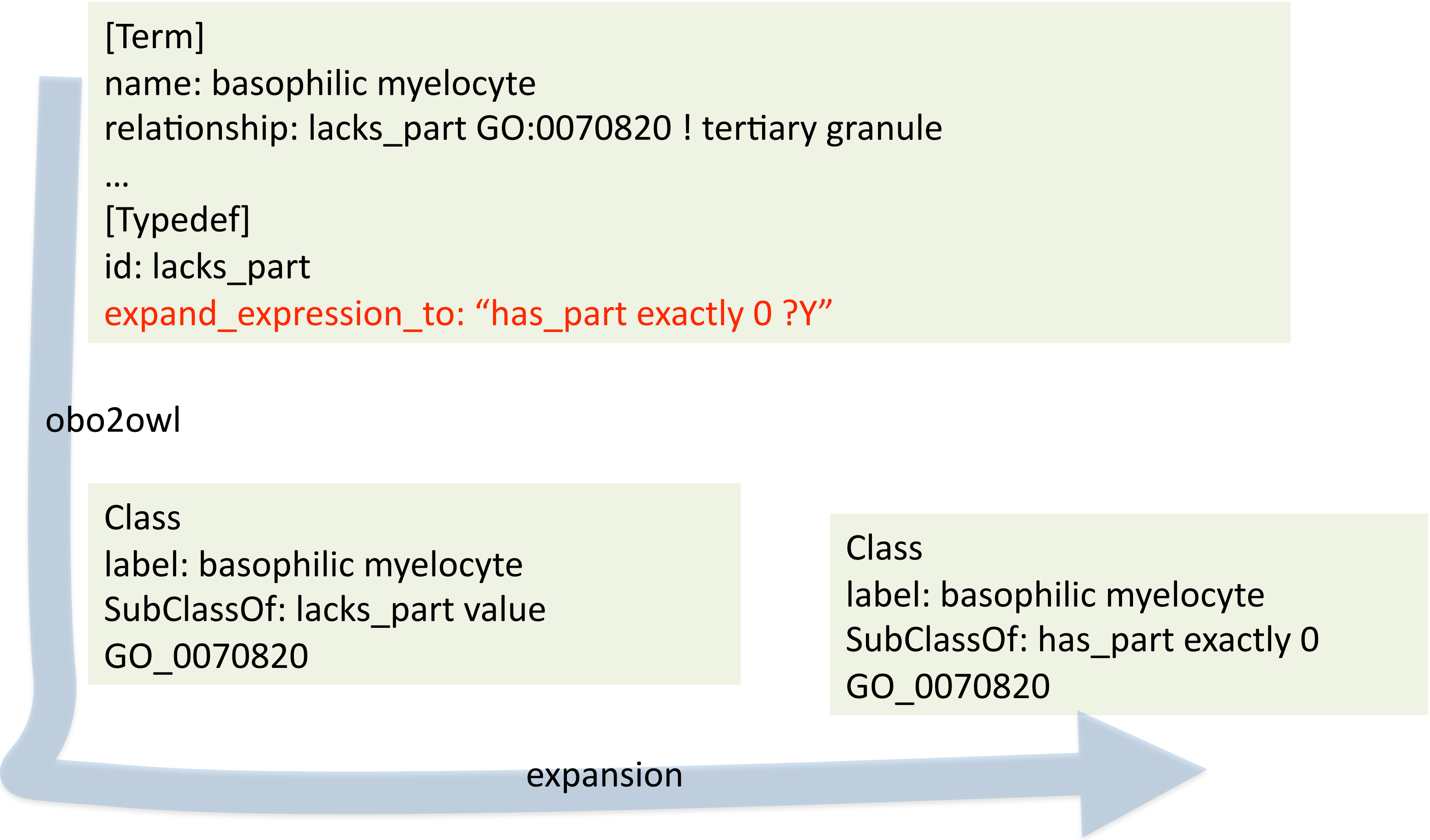
You can have your cake and eat it

Smuggling OWL
into obo-format

```
[Typedef]
id: has_plasma_membrane_part
name: has_plasma_membrane_part
is_a: has_part
expand_expression_to: “has_part_some (GO:0005886 and has_part some ?Y)”
```

instructions on how to expand the
shortcut relation

Does all this make a difference? YES: A real life example



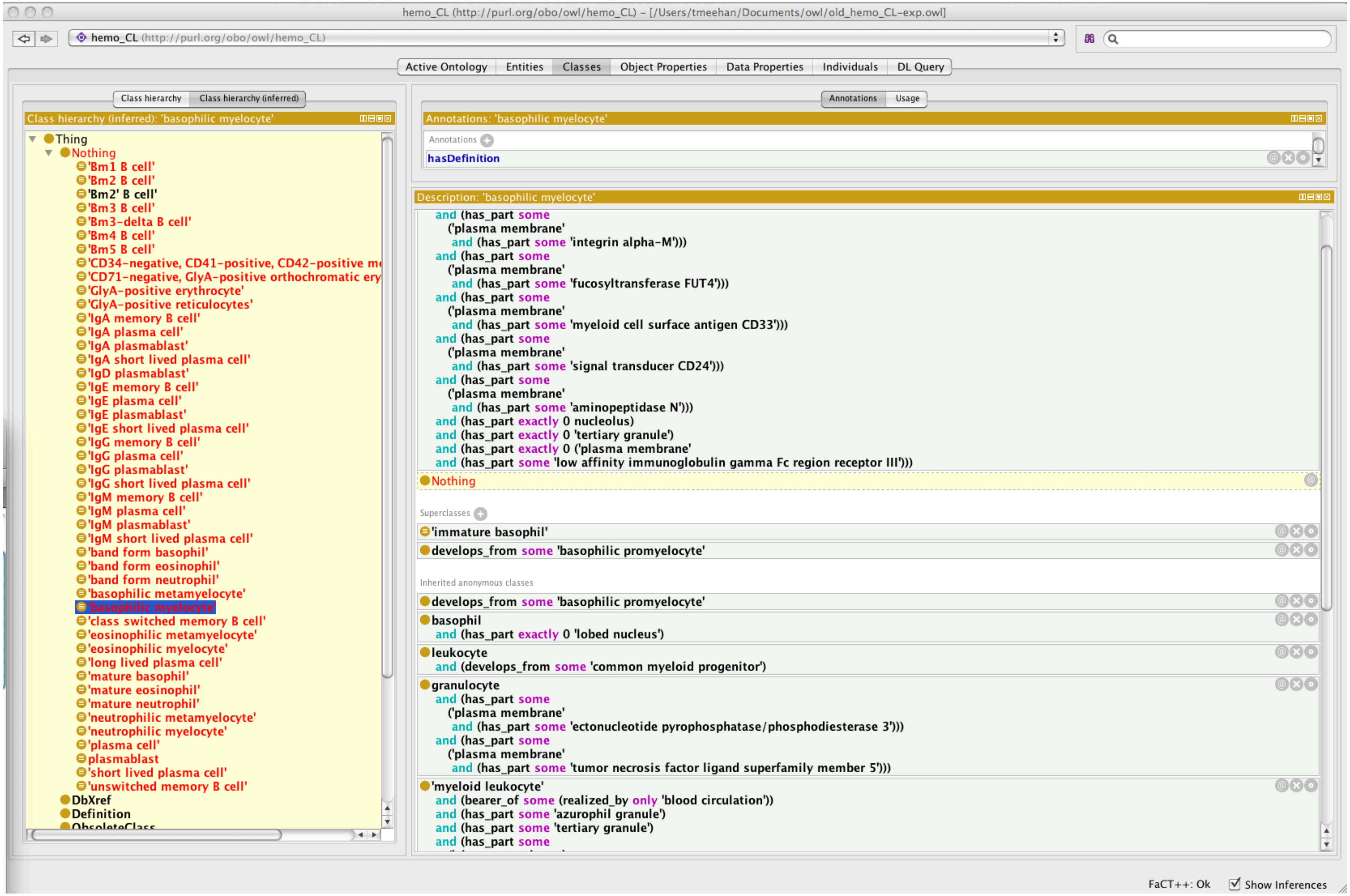
```
[Term]
name: basophilic myelocyte
relationship: lacks_part GO:0070820 ! tertiary granule
…
[Typedef]
id: lacks_part
expand_expression_to: “has_part exactly 0 ?Y”
```

obo2owl

```
Class
label: basophilic myelocyte
SubClassOf: lacks_part value
GO_0070820
```

```
Class
label: basophilic myelocyte
SubClassOf: has_part exactly 0
GO_0070820
```

expansion


hemo_CL (http://purl.org/obo/owl/hemo_CL) – [/Users/tmeehan/Documents/owl/old_hemo_CL-exp.owl]
hemo_CL (http://purl.org/obo/owl/hemo_CL)
Active Ontology
Entities
Classes
Object Properties
Data Properties
Individuals
DL Query
Class hierarchy
Class hierarchy (inferred)
Annotations
Usage
FaCT++: Ok
Show Inferences

Class hierarchy (inferred): 'basophilic myelocyte'

```
Thing
  Nothing
    'Bm1 B cell'
    'Bm2 B cell'
    'Bm2' B cell'
    'Bm3 B cell'
    'Bm3-delta B cell'
    'Bm4 B cell'
    'Bm5 B cell'
    'CD34-negative, CD41-positive, CD42-positive m
    'CD71-negative, GlyA-positive orthochromatic ery
    'GlyA-positive erythrocyte'
    'GlyA-positive reticulocytes'
    'IgA memory B cell'
    'IgA plasma cell'
    'IgA plasmablast'
    'IgA short lived plasma cell'
    'IgD plasmablast'
    'IgE memory B cell'
    'IgE plasma cell'
    'IgE plasmablast'
    'IgE short lived plasma cell'
    'IgG memory B cell'
    'IgG plasma cell'
    'IgG plasmablast'
    'IgG short lived plasma cell'
    'IgM memory B cell'
    'IgM plasma cell'
    'IgM plasmablast'
    'IgM short lived plasma cell'
    'band form basophil'
    'band form eosinophil'
    'band form neutrophil'
    'basophilic metamyelocyte'
    'basophilic myelocyte'
    'class switched memory B cell'
    'eosinophilic metamyelocyte'
    'eosinophilic myelocyte'
    'long lived plasma cell'
    'mature basophil'
    'mature eosinophil'
    'mature neutrophil'
    'neutrophilic metamyelocyte'
    'neutrophilic myelocyte'
    'plasma cell'
    plasmablast
    'short lived plasma cell'
    'unswitched memory B cell'
  DbXref
  Definition
  ObsoleteClass
```

Annotations: 'basophilic myelocyte'

Annotations

hasDefinition

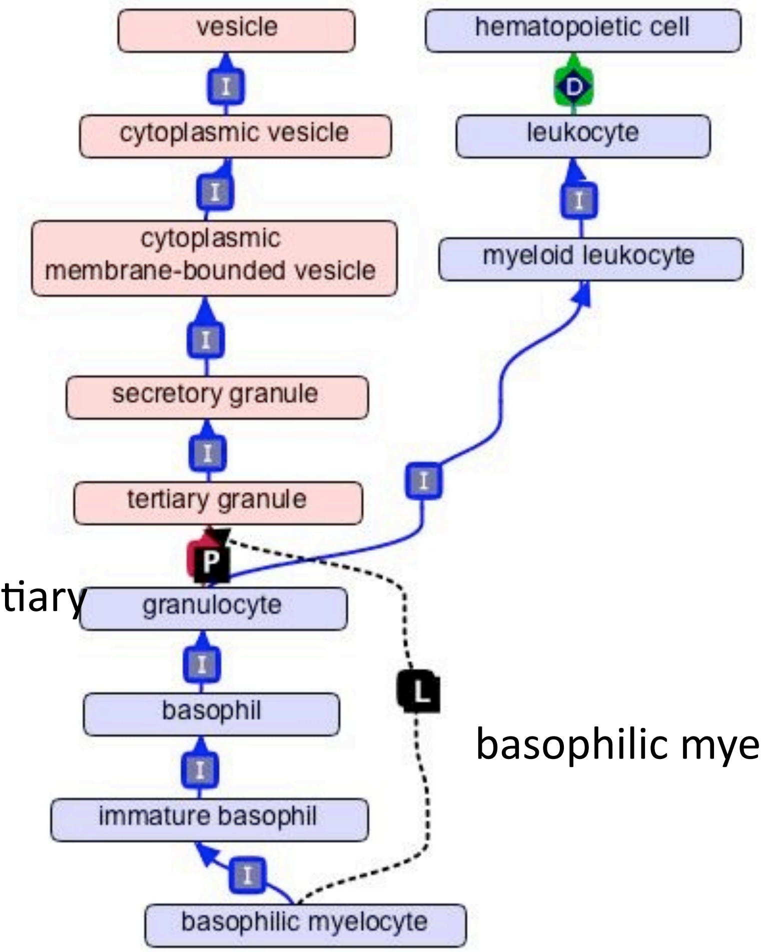
Description: 'basophilic myelocyte'

```
  and (has_part some
    ('plasma membrane'
     and (has_part some 'integrin alpha-M')))
  and (has_part some
    ('plasma membrane'
     and (has_part some 'fucosyltransferase FUT4')))
  and (has_part some
    ('plasma membrane'
     and (has_part some 'myeloid cell surface antigen CD33')))
  and (has_part some
    ('plasma membrane'
     and (has_part some 'signal transducer CD24')))
  and (has_part some
    ('plasma membrane'
     and (has_part some 'aminopeptidase N')))
  and (has_part exactly 0 nucleolus)
  and (has_part exactly 0 'tertiary granule')
  and (has_part exactly 0 ('plasma membrane'
  and (has_part some 'low affinity immunoglobulin gamma Fc region receptor III')))
Nothing

Superclasses
'immature basophil'
develops_from some 'basophilic promyelocyte'

Inherited anonymous classes
develops_from some 'basophilic promyelocyte'
basophil
  and (has_part exactly 0 'lobed nucleus')
leukocyte
  and (develops_from some 'common myeloid progenitor')
granulocyte
  and (has_part some
    ('plasma membrane'
     and (has_part some 'ectonucleotide pyrophosphatase/phosphodiesterase 3')))
  and (has_part some
    ('plasma membrane'
     and (has_part some 'tumor necrosis factor ligand superfamily member 5')))
'myeloid leukocyte'
  and (bearer_of some (realized_by only 'blood circulation'))
  and (has_part some 'azurophil granule')
  and (has_part some 'tertiary granule')
  and (has_part some
```

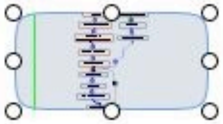
hemo CL inconsistency



granulocyte *has_part* tertiary granule

OE reasoner does not detect this inconsistency (it doesn’t know that lacks part and has part conflict)

basophilic myelocyte *lacks_part* tertiary granule



Status

- Current implementation is prototype
  - Being used for hemo_CL
  - Will be rewritten to use OWLAPI
- Awaiting comments from OWL community
  - Could be useful for OWL/Protégé4 community in general

Alternatives

- OWL allows relation chains
  - Some expressible in OBO
  - E.g.
    - X has_part M, M is_a plasma_membrane, M has_part Y → X has_plasma_membrane_part Y
- Not sufficient for our purposes

Summary of shortcut formalism

- Shortcut relations
  - easier for non-OWL heads to understand
  - save on repetitive strain injury
  - easier to visualize
  - can be consumed by basic ontology applications and databases
- Expanded form
  - machine code for reasoners
  - can be hidden from users (even in P4)
  - purists may prefer this

Another application: Connecting cells to GO BP

- we want to be able to say something like "a function of the osteoclast is bone resorption $_{\text{GO:0045453}}$". How?
  - osteoclast participates_in GO:0045453
    - wrong – not every osteoclast is doing this
  - osteoclast has_function GO:0045453
    - sorry - GO:0045453 is not a BFO function!
  - We could make a biological function ontology paralleling GO, and use this
    - osteoclast has_function GO_BF:to_resorb_bone
    - we *could*, I guess, but that would be madness

Solution: use a shortcut

- OWL and BFO purist way:
  - osteoclast bearer_of some (bfo:function and realized_by only GO:bone resorption)
    - or:
  - osteoclast bearer_of some (bfo:function and realized_by only (part_of some GO:bone resorption))
  - but who wants to write this all the time?
- Shortcut
  - capable_of → bearer_of some (realized_by only ?Y)

Summary: capable_of

- This is the relation the majority of users of the ontology would see
- Optional expansion in OWL
  - – not clear the extent to which expansion will help with reasoning
    - Reasoning with unexpanded form in obo and owl may be fine
- Open question
  - – do we need sub-relations for functions vs roles

Comparison with fly_anatomy

- uses has_function_in
- slightly more specific (the cell must have a function realized by the process, rather than just play a role)
- in David's talk?
- No strong use case for expanding the relation thus far

Ongoing issues

- How do we expand:
  - has_high_plasma_membrane_amount
  - has_low_plasma_membrane_amount
- Use populations?
  - ➔ has_part some (population and has_grain ?Y and larger_than ref_pop_0001124)
  - unwieldy, still doesn’t capture full semantics
  - perhaps fine to leave as documented but unexpanded for now
- We can express:
  - has_low_plasma_membrane_amount disjointFrom lacks_part

Summary

- Shortcut relations proving useful for hemo_CL
  - Also being used in
    - fly_anatomy (neurons) -- later today
    - NIF cell->anatomy
- Ontology authors and biologist users focus on high level shortcut relations
  - division of labor

- END

Sensu

The demise of sensu in GO

- GO 1999ish
  - cell wall (sensu fungi)
  - cell wall (sensu plant)
- GO post 2006
  - cell wall, beta-glucan/chitin
  - cell wall, cellulose and pectin

  - Differentia based on **non-taxonomic** criteria
    - e.g. structural

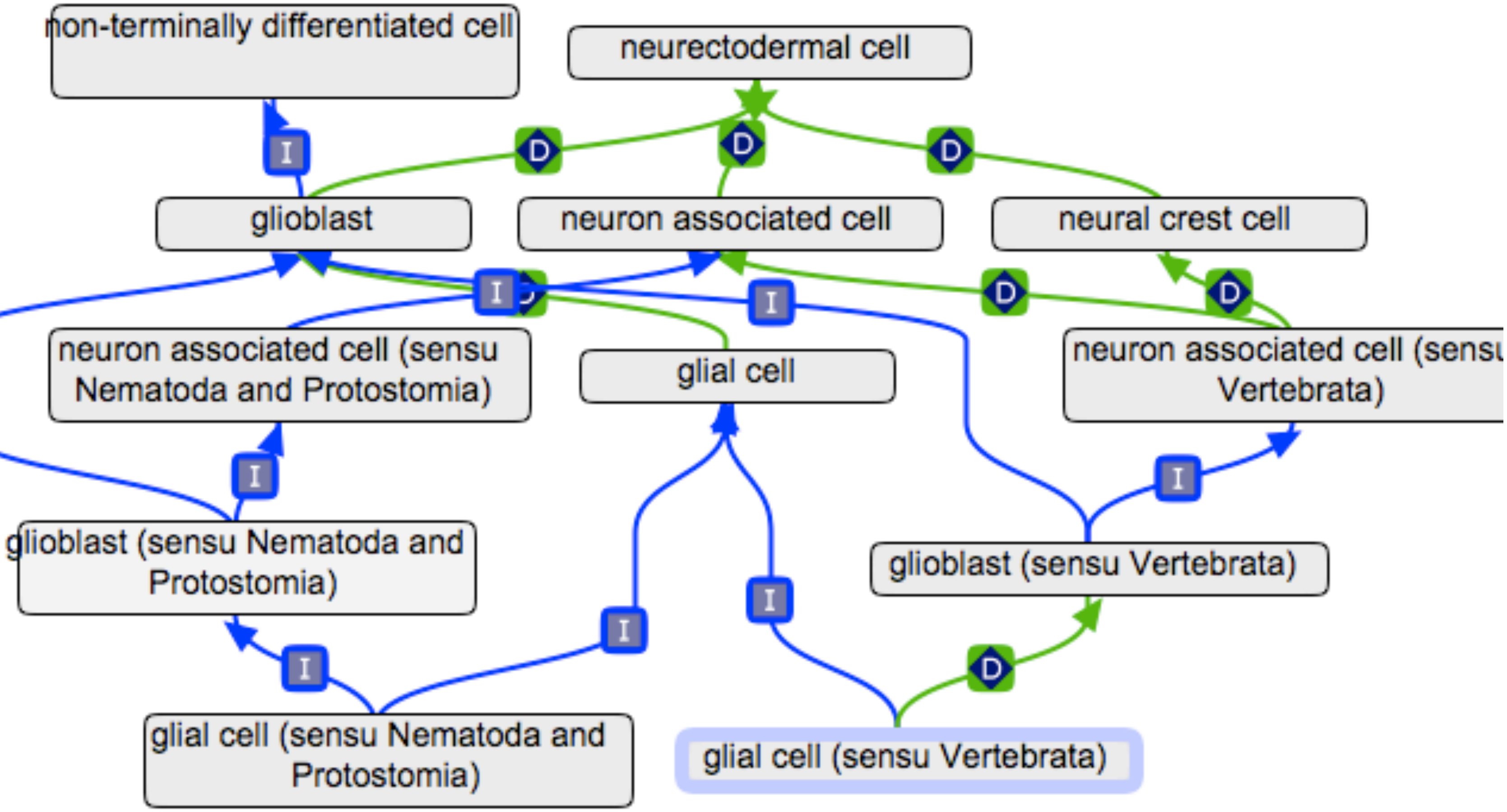
Taxa in GO

- only_in_taxon
  - mammary gland development only_in_taxon Mammalia
  - glial cell differentiation only_in_taxon Metazoa
- never_in_taxon
  - odontogenesis never_in_taxon Aves

Fixing CL

- Original information on why sensu qualifiers added not in ontology

Example


non-terminally differentiated cell
neurectodermal cell
glioblast
neuron associated cell
neural crest cell
neuron associated cell (sensu Nematoda and Protostomia)
glial cell
neuron associated cell (sensu Vertebrata)
glioblast (sensu Nematoda and Protostomia)
glioblast (sensu Vertebrata)
glial cell (sensu Nematoda and Protostomia)
glial cell (sensu Vertebrata)
I
D

CL in AOs

- The following AOs include representations of cell types
  - Fly anatomy (FBbt) (includes CL xrefs)
  - Zebrafish anatomy (ZFA) (includes CL xrefs)
  - Adult human (FMA)
  - Plant (PO)
- Mouse anatomy (MA) does not
- Why replicate?
  - Relationships to gross-anatomy
  - ‘Too specific’ for CL

Current approach

- Some AOs maintain xrefs to CL
- Sometimes CL has xrefs to AOs
  - E.g. FMA
- We treat the ss AOs classes as subclasses
  - not ideal

Proposed approach

- No duplication
  - OBO-Foundry ontologies MIREOT CL
  - Reuse classes
    - ss AOs can include cell types that are genuinely only characterized in that species
  - Exception: PO?
- Relationships to AO of appropriate level of specificity

- END

Use of CL in GO

using CL in the GO

- Ontology authoring
  - CL classes used in logical definitions
    - astrocyte differentiation
- Annotation
  - Post-composition

CL in GO pre-composed classes

- Bridge files
  - biological_process_xp_cell (603)
  - cellular_component_xp_cell (26)
  - cellular_component_links_cell (28)
- Reasoning
  - pre-ARRA
    - multiple inconsistencies (CL tracker, list)
  - post-ARRA
    - asserted links in GO now match inferred links

Post-composition

- ‘column 16’
  - E.g. occurs_in(CL:0001234)
  - Formal semantics in gene associations
    - <col5> and R some Y
- Already in use
  - MGI
  - Pombe
    - (but not using CL)

Challenges with post-composition

- Reconciling pre- and post-
  - Scenario:
    - Joe annotates to apoptosis & occurs_in(cardiac cell)
    - GO introduces pre-composed “cardiac cell apoptosis”
    - We want to include Joe’s annotation here
  - Solution: solved using standard reasoning techniques
- Term enrichment, semantic similarity
  - Term enrichment is one of the main uses of GO
  - Can CL enhance this?
  - Can use LCS techniques developed by DL researchers (1990s-present)
    - Algorithms solved (for certain cases)
    - implementations lacking!

Timelines

- Phase I
  - concentrate on pre-composition
  - working with annotators to explain post-composition
- Phase II
  - Full support of post-composition in AmiGO

- END