# GO Software Group
# - Progress and Next 5 years

*working more efficiently for you*

# Theme

- 2000-2010
  - "wild west"
  - home-grown formats and tools
  - development of de-novo software
- 2010-2020
  - mature phase
  - 3rd party tools
  - Software group as *integrators*
  - Increased Automation

# Outline

- **Support for Ontology Development**
  - TermGenie
  - Leverage OWL tools
- **Annotation and Reference Genome Support**
  - QC and Rule Engine
  - Expressivity and automatic integration
  - Annotation Tools
- **Web Presence**
  - AmiGO and QuickGO
  - Galaxy
- **Infrastructure and GO Database**
  - Database Overhaul
  - Virtualization and the cloud

# Ontology Development

**First Decade**

- Large monolithic, manually constructed graph
- Sourceforge workflow
- All ontology changes through editor + OboEdit

**Second Decade**

- Modular ontology construction
  - 'outsourcing' to OBO
  - MIREOTing of terms
  - automated classification
- Instant Compositional Terms for Annotators
  - TermGenie
- Reasoner-based QC

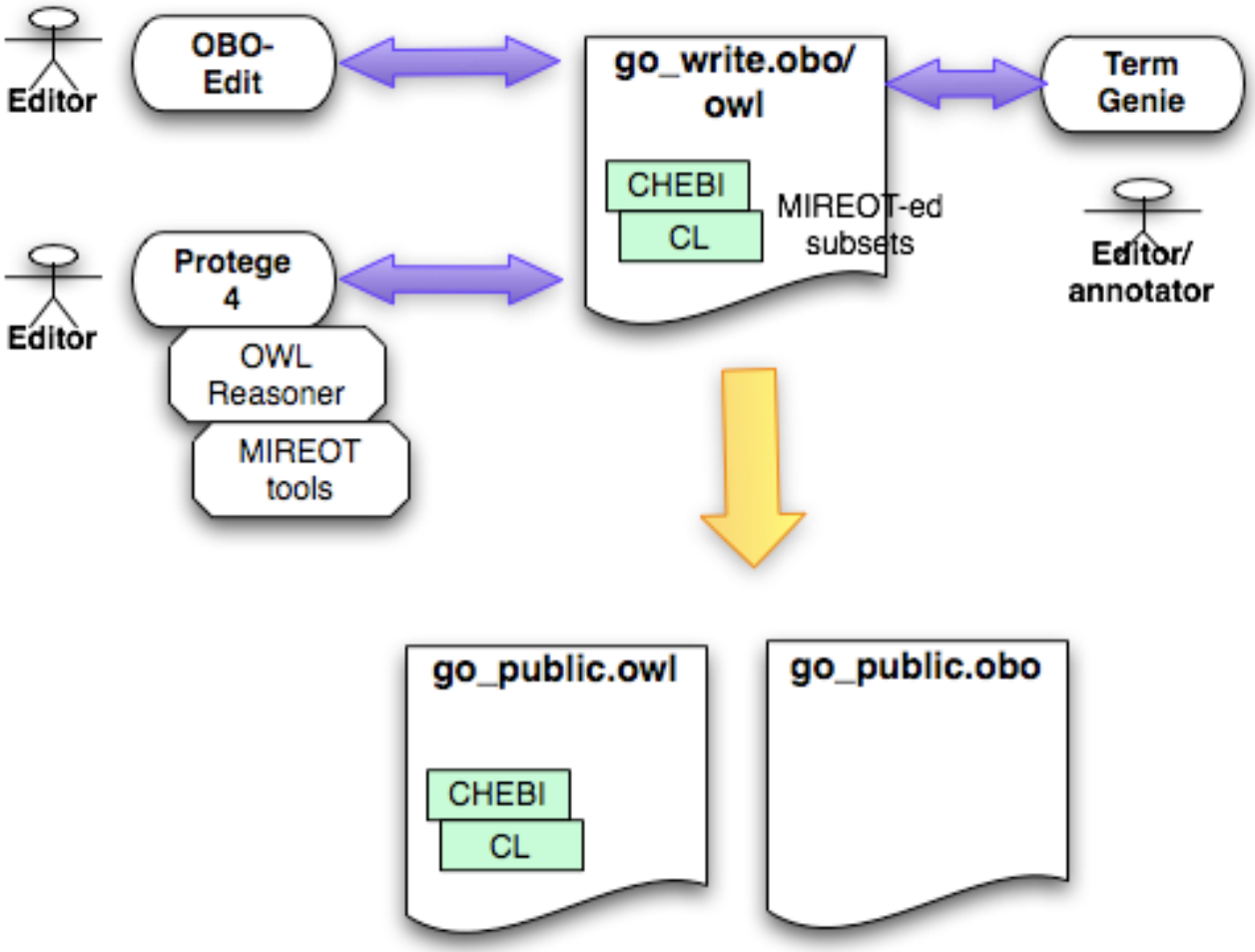
# Current progress: Ont-Dev

- TermGenie
- Support for MIREOT/isa closure in OboEdit
- More QC checks

# Infrastructural changes required to support ontology development

- We have too much dependence on home grown software
  - reasoning, ontology processing and editing
  - poor bus ratio
- Most useful 3rd party ontology development tools assume OWL
  - We will make obo-format1.4 formally correspond to a subset of OWL
    - Write reliable converters
  - Migrate all code to OWL API

# Workflow


Editor
OBO-Edit
go_write.obo/owl
Term Genie
CHEBI
CL
MIREOT-ed subsets
Editor/ annotator
Editor
Protege 4
OWL Reasoner
MIREOT tools
go_public.owl
CHEBI
CL
go_public.obo

# SWUG Plan: Ont-Dev

- Freeze OE new features
  - maintenance mode
- Prioritize obo/owl conversion
  - allows people to use whichever tool is most appropriate
- Make OE3 a plugin for Protege4
  - port visualization, verification checks
- Migrate existing ont support tools to OWLAPI
  - TermGenie
  - OE Reasoner
  - Ontology QC reports

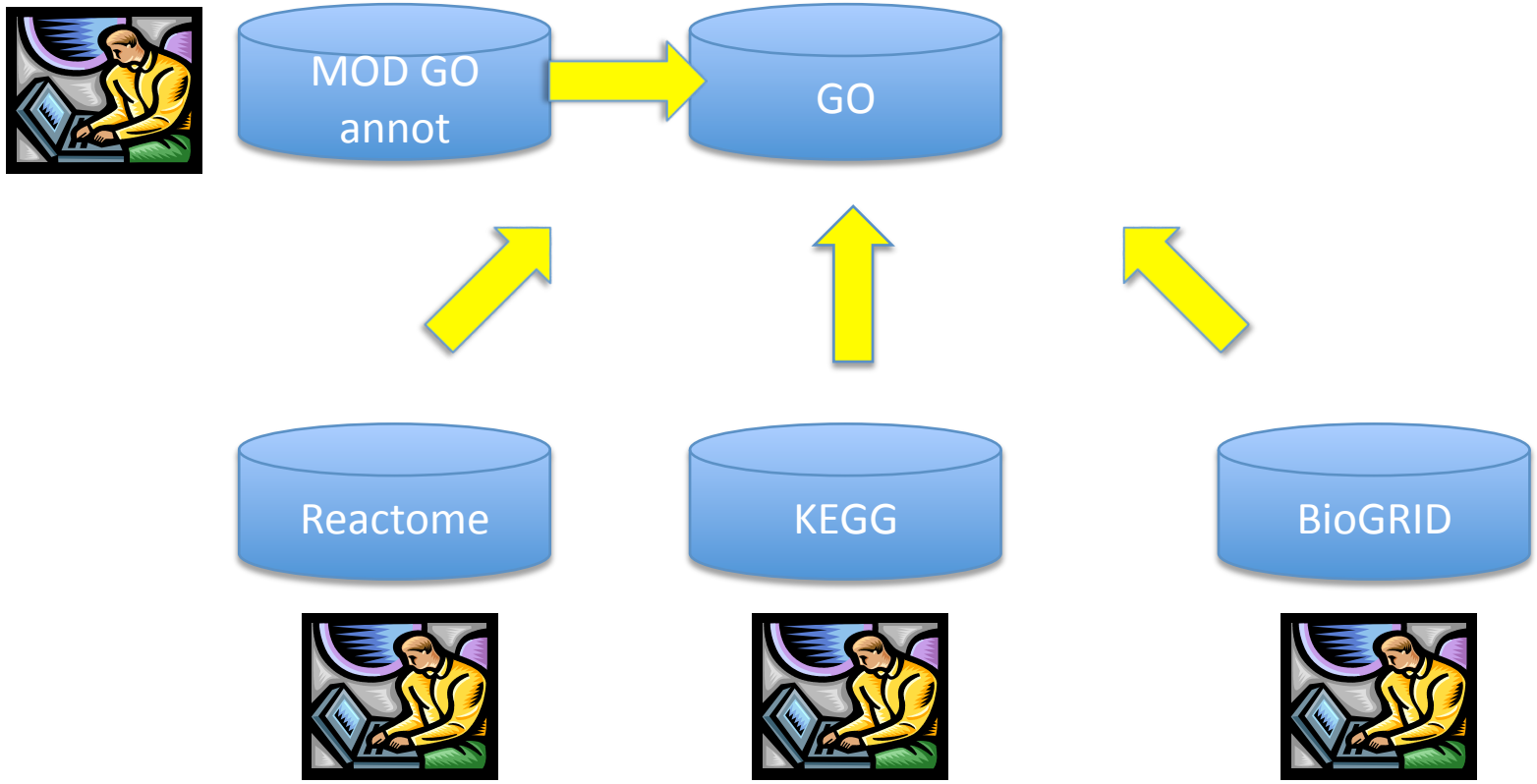
# Annotation support

- Automated QC and inference
- Automatic integration with external databases
- Support increased expressivity
- Annotation Tools

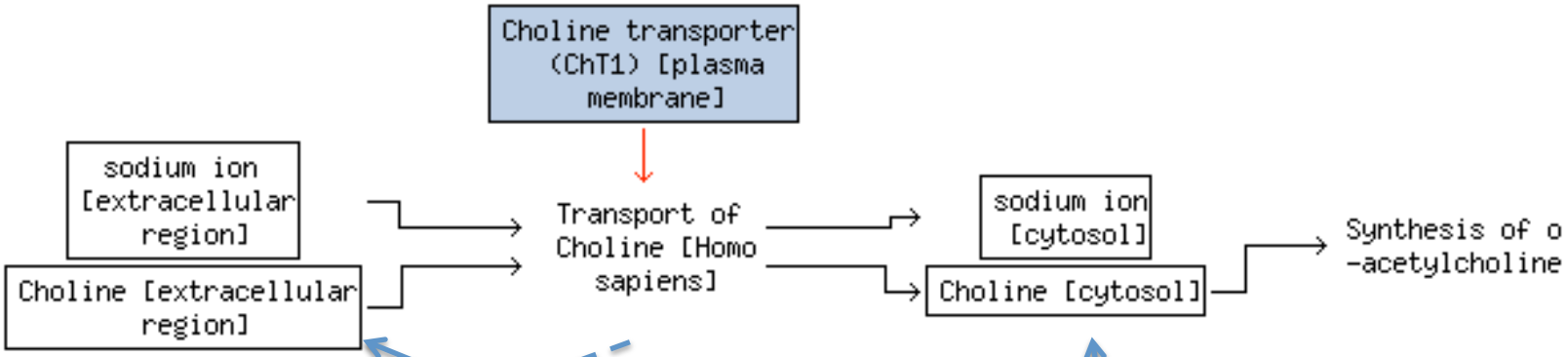
# Automatic QC and inference

- Existing checks and inferences:
  - Taxon constraints
    - paper accepted for BMC bioinformatics
  - Materializing F->P annotations
  - Ad-hoc SQL queries and scripts
- Plan
  - Unified **Rule Engine**
  - Driven from central rules file
  - Implemented using OWL API

# Automatic integration



# Automating integration using computable definitions – **pathway databases**

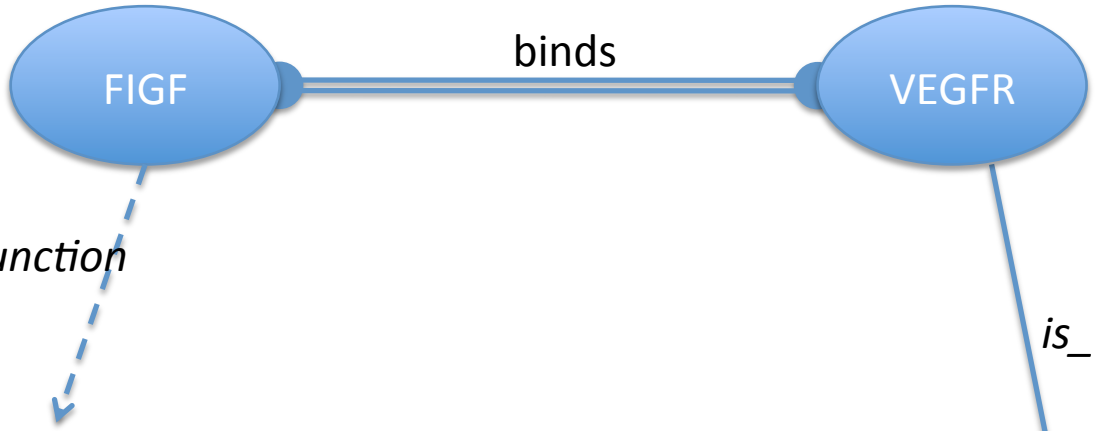



```
[Term]
id: GO:0015871
name: choline transport
intersection_of: GO:0006810 ! transport
intersection_of: results_in_transport_of CHEBI:15354 ! choline
```

**Implementation**: standard reasoning techniques

**TODO**: port to OWL API

# Interaction dbs / binding




```
[Term]
id: GO:0043184
name: vascular endothelial growth factor receptor 2 binding
intersection_of: GO:0005488              ! binding
intersection_of: results_in_binding_of PRO:000002112 ! VEGFR 2
```

**Implementation**: standard reasoning techniques
**TODO**: port to OWL API

# Increasing expressivity

- Current
  - col16
    - other ontologies
    - gene product targets – “mini pathway” annotations
  - col17
- Increasing coverage of expressive annotations
  - pull automatically from pathway databases
- Relationship between col16 and LEGO
- Integral_to qualifier
- Still to do
  - Automatically deepening annotations
    - standard reasoning technology
  - Using col16 in term enrichment

# Annotation Tools

- Current
  - PAINT
    - phylogenetic inference
    - desktop application
  - Individual MOD annotation interfaces
- Future
  - Web PAINT
  - Common Web Annotation Interface “IndiGO”
    - curators and community
    - reuse: p2go, textpresso, phenote, ..

# Reference Genome Tracking and Reporting

- Database Reports
- More pro-active use of wikis and collaborative tools
  - integrate GONUTs and GO Wiki
  - Sourceforge replacement

# Derived Metrics

- How do evaluate how we're doing?
  - Systematically evaluate

- Example
  - Does integrating with pathway dbs help?
    - Let's take: Genes down-regulated in Alzheimer's

| | GOA without *R* | GOA with *R* (enhanced) |
|---|---|---|
| oxidative phosophorylation | $7 \times 10^{-29}$ | $1.2 \times 10^{-44}$ |
| regulation of insulin secretion | 0.72 | $4 \times 10^{-46}$ |

# AmiGO 1.8 Progress

- Available on labs
  - improved lucene-based search
  - web services for visualization
  - advanced queries
    - N-matrix (see Val's talk from GO annot camp)
  - new term pages

  - http://amigo.berkeleybop.org/cgi-bin/amigo/term_details?term=GO:0022008

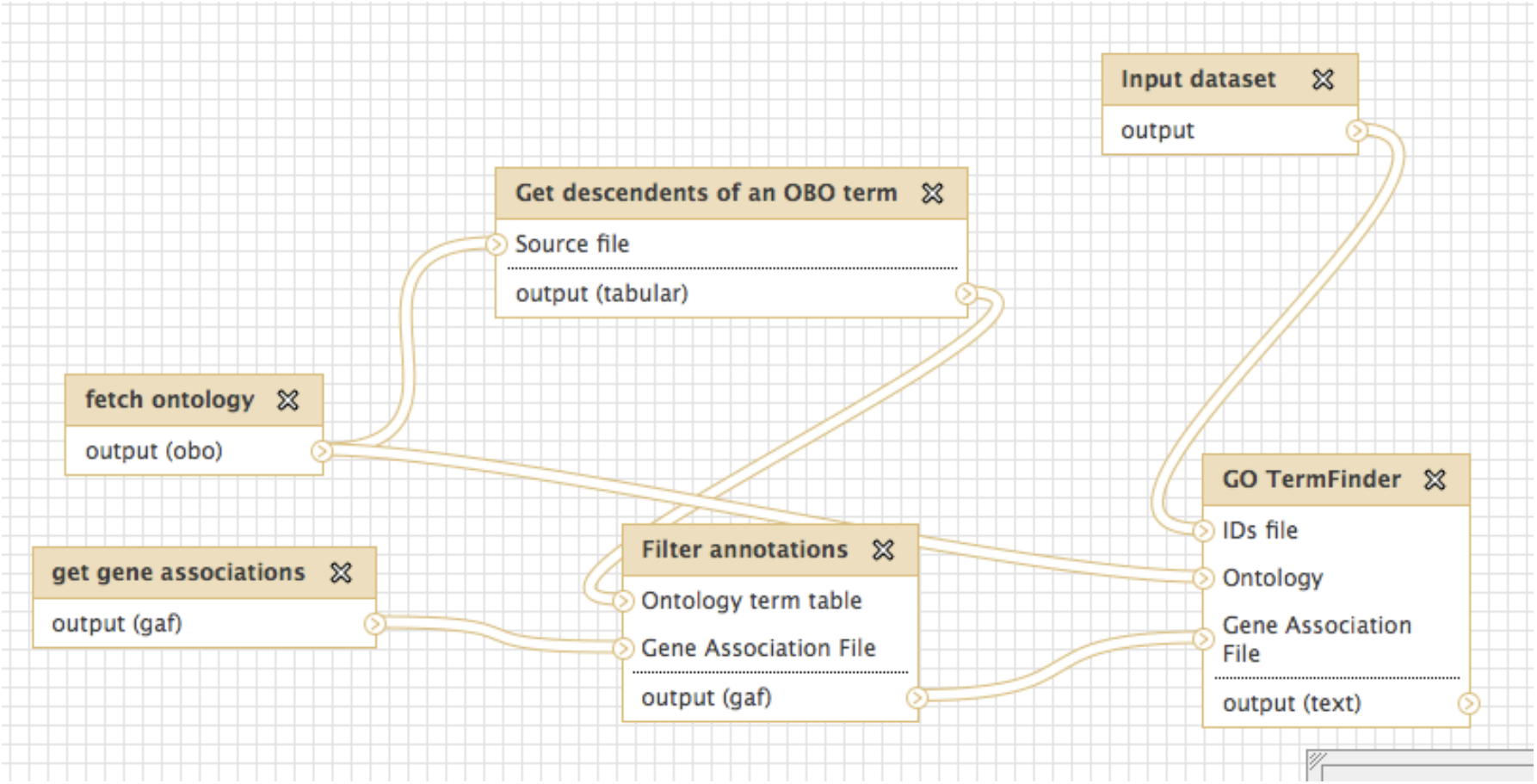
# AmiGO and QuickGO

- Overlapping core functionality
  - duplicate code
  - wasted effort
- Current strategy
  - Loose coupling
  - AmiGO labs now shows quickgo graphs
- Future strategy
  - tighter integration
  - shared java codebase

# GO Tools

- We list >50 on website
  - We have started capturing more detailed metadata
- Lots of effort for users
- Requires bioinformatics expertise to build workflows
  - ID mapping
  - mapping using orthologs
  - mapping using ext2go
  - building and using slims for analyses
- Current progress:
  - shopping carts in AmiGO

# GO Galaxy Environment




http://berkeleybop.org/galaxy

# Database and Infrastucture

- Future of GO database
- Deployment and virtualization

# Future of GO Database

- **Currently used for many different purposes:**
  - underpins AmiGO
  - underpins PAINT
  - SQL Queries for Annotation QC checks
  - advanced user GOOSE queries
  - mirrored internally by a number of groups

# Future of GO Database

- **Currently used for many different purposes:**
  - underpins AmiGO
  - underpins PAINT
  - SQL Queries for Annotation QC checks
  - advanced user GOOSE queries
  - mirrored internally by a number of groups
- **…But there are problems:**
  - designed in 1999
  - inefficient for querying and bulkloading
  - mysql
  - outdated perl middleware

# Alternatives to RDBMSs

- Text indexing engines
  - Apache Lucene/SOLR
- Key-value databases
  - Google BigTable
- RDF Triplestores
  - ontology-aware SPARQL queries
- In-memory querying
  - OWL API
- Custom indexing
  - QuickGO

# Database Strategy

- AmiGO queries
  - Use Lucene/SOLR
  - In-memory ontology querying
- Other GO functions
  - Resdesign/simplify relational schema
    - (ARRA)
- In parallel
  - Leverage external RDF stores
    - Neurocommons

# Virtualization and the cloud

- Current deployment cycle is inefficient
- Solution: virtual machine (VM) images
  - database(s)
  - amigo/quickgo
  - GO annotation tools + galaxy server
  - piggyback off existing genomics VM
- Can be easily deployed on a variety of servers or on the cloud

# Summary

- Key areas
  - ontology and annotation automation and integration
- Increased efficiency
  - reuse
    - OWL API
  - collaborate
  - lightweight