InterMine and Chado

Richard Smith
University of Cambridge
Overview

- Query-optimised data warehouse system
- Java, object-based data model
- Flexible querying

Diagram:

- Integrate data
- InterMine data warehouse
- Web application
- Query API
- Web services
Projects

- FlyMine - www.flymine.org
  - 30+ data sources, *Drosophila* & *Anopheles*
- modENCODE - www.modencode.org
  - *C. elegans/D. melanogaster* high throughput
- BOKU & IMP - Vienna
- MitoMiner - mitochondria
- MilkMine - milk proteins
- Model organisms
Adapt performance to actual usage +
Performance optimisation separate from schema design

Architecture

Web Application
- Java query
- Java objects

ObjectStore (O/R mapping)
- generated SQL
- optimised SQL

Query Optimiser
- Queries optimised to join fewer tables and execute faster

PostgreSQL
- precomputed tables
- master tables

Object cache
- objects are cached by id for fast access
Data Integration

- UniProt XML
- e!
- Chado
- OMIM disease
- OBO ontology
- PSI XML
- Inparanoid
- GO

Custom Data Sources
- FASTA
- GFF3
- XML

Java and Perl APIs

Configurable data integration

InterMine data warehouse
Auto-generation

- Low overhead to extending data model

- Core model
  + Model additions
  + Model additions

- Data Model

- Database schema
- Java classes
- Web application
Web Application

- Works for any data model
- Highly configurable
- Configuration from within web interface
Webapp Overview

- **Query Builder**
  - Constrain attributes & select fields
  - Add / Rearrange columns, Modify Query
  - Build template / Return to Query Builder

- **Template Library**
  - Use list with Template Query or Query Builder
  - List conversion/set operations with other lists

- **Results Table**
  - Select items

- **List**

- **Report Page**

- **Quick Search**

- **Export:**
  - GFF
  - Tab Delimited
  - FASTA
  - Excel

- **List Analysis Page**

- **Upload List**
Demo
Chado sources

- FlyBase - genome, publications, alleles
- WormBase - genome only
- modENCODE DCC submissions
Import any Chado

- Periodic import of Chado data
- Work for any Chado instance
- Default templates and widgets
Data model

- Java class per SO term
- Inheritance to represent is_a relationships
- part_of -> Java references/collections
- featureprops -> attributes
- Similar approach to Chado for locations
Data Loading

**ChadoSequenceProcessor**
- reads: sequence module + dependencies
- creates: features, locations, synonyms, pubs
- sets: identifiers as attributes
- sets: references/collections from part_of feature_relationships

**modENCODEProcessor**
- DCC metadata

**FlyBaseProcessor**
- e.g. alleles
- e.g. cyto_range

**WormBaseProcessor**
Possible improvements

• Cleaner Java API
• Java code -> configuration file
• Point and click setup
• Default templates and widgets
• Derive data model automatically
Acknowledgments

**Biologists** Hilde Jannsens, Rachel Lyne

**Developers** Richard Smith, Jakub Kulaviak, Julie Sullivan, Matthew Wakeling, Xavier Watkins

**Sys Admin** Dan Tomlinson

**modENCODE** Sergio Contrino, Kim Rutherford

**PI** Gos Micklem