



# OpenFlyData: the way to go for biological data integration

Dr Jun Zhao
Image Bioinformatics Research Group
Department of Zoology
University of Oxford

# OpenFlyData Application

- To answer questions about
  - "what does this gene do?”
  - “which genes are of interests?”
  - ….
- Investigate the feasibility of existing Semantic Web tools and technologies for real applications
- Create a set of reusable data sources and data query services

# mRNA gene expression study

- Microarray analysis
  - How much of a given transcript (mRNA) is present in a sample
  - In a quantitative way
  - Lack of spatial information
- RNA *in situ* hybridization
  - Reveal both spatial and temporal aspects of gene expression during the development
  - But not quantitative

http://www.fly-ted.org.uk/509/

# Use cases of gene expression data

- Experimental design
  - Choosing which handful of genes/mutations to study in detail
- Data validation
  - Mitigating problems of cross contamination

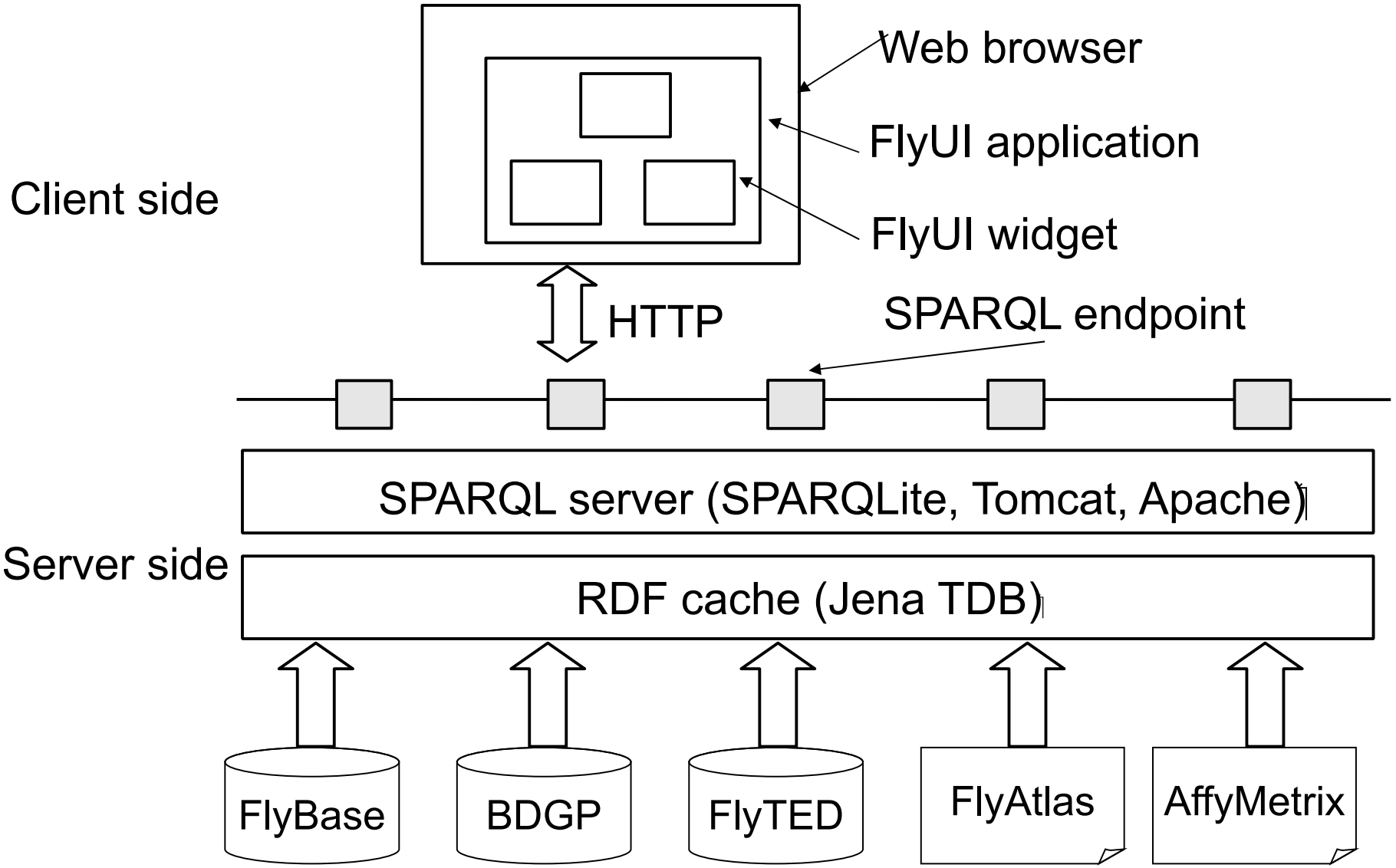
# Barriers for accessing these data

- Data are scattered at different web sites
- Searches have to be repeated, different search interfaces, different use of terminology
- Limited (if any) programmatic access to data … hard work to answer questions that span data sources

# OpenFlyData.org demonstration

- Three gene express cross-database search applications
  - Search by gene, gene expression mashup: [go]
  - Search gene expression by gene batch [go]
  - Search gene expression by tissue expression profile [go]

# System architecture



# SPARQL queries

SPARQL

```
PREFIX chado: <http://purl.org/net/chado/schema>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xs: <http://www.w3.org/2001/XML_Schema#>
SELECT ?flybaseID
WHERE {
    ?feature rdf:type chado:Feature ;
             chado:name "schuy"^^xs:string ;
             chado:uniquename ?flybaseID .
}
```

SQL

```
SELECT ?feature.uniquename AS flybaseID
FROM feature
WHERE feature.name = "schuy"
```

# SPARQL protocol

HTTP GET

```
GET /query/flybase?query=[URL encoded query] HTTP/1.1
Host: openflydata.org
Accept: application/sparql-results+json
```

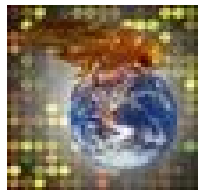
HTTP POST

```
POST /query/flybase HTTP/1.1
Host: openflydata.org
Accept: application/sparql-results+json
Content-Type: application/x-www-form-urlencoded
Content-Length: 456
query=[URL encoded query]
```

# The data sources

- Flybase and BDGP: GMOD relational databases
- FlyTED, an image repository built using Eprints
- FlyAtlas, tissue-specific *Drosophila* gene expression levels, as a single spreadsheet



# Creating RDF from data sources

- D2RQ mapping
  - FlyBase and BDGP, native relational databases
  - Conservative mapping, with minimum interpretation
- OAI2SPARQL
  - Harvesting N3 RDF metadata via the OAI-PMH protocol, built-in support by Eprints
  - Further from ESWC2008 paper
- Custom Python program
  - FlyAtlas
  - Generating N3 from spreadsheet table

# The heterogeneous Drosophila gene names

| DATA SOURCE | POSSIBLE GENE IDENTIFIERS | EXAMPLES |
|---|---|---|
| FlyBase | symbol | schuy |
| | full name | schumacher-levy |
| | annotation symbol | CG17736 |
| | Unique FlyBase id | FBgn0036925 |
| | Curated synonyms | CG17736, schuy, etc |
| BDGP | FlyBase id | FBgn0036925 |
| | Annotation symbol | CG17736 |
| FlyAtlas | Affy microarray probe id | 16166608_a_at |
| FlyTED | Uncontrolled gene name | schuy, CG17736/schuy |

# Gene name mapping

- Use FlyBase for automatic gene mapping
  - 67 out 833 genes remain unmapped or ambiguously mapped
- Additional inputs from scientists for disambiguating many-many mappings
  - Only one gene remains unmapped
- Mappings are stored as JSON file to assist "GeneFinder" widget (having no use for RDF/OWL reasoning at this stage)
- Evaluation of gene name mapping
  - Test the SPARQL queries to the FlyTED store
  - Test the FlyTED gene expression search service

# SPARQL server

- Amazon EC2 (Elastic Compute Cloud):
  - To run SPARQL endpoints
  - To host the demo you've just seen
- Jena TDB as triple store
  - For better loading performance: ~6K tps for ~9M triples to Amazon Elastic Block Storage (EBS)
  - For better querying performance
- SPARQLite
  - home-grown SPARQL protocol implementation
  - More later
- Apache, Tomcat, mod_jk, etc.

# SPARQLite protocol

- http://sparqlite.googlecode.com
  - A platform for exploring SPARQL service quality concerns, more later
  - Restricted forms of query (SELECT, ASK and DESCRIBE)
  - Restricted query result format (e.g. only JSON)
  - Disallow queries containing variable predicates.
  - Disallow queries containing a “filter” clause and/or using unbound variables.
  - Limit the maximum number of requests sent from any one source in any one second to 5
- Designed for Jena TDB/SDB + Postgres

# Benefits of SW technologies

- RDF provides a uniform and flexible data model
  - RDF dump is cheaper and quicker
  - Maintaining a separate SPARQL endpoint for each data source makes it easier than a data warehouse approach for handling data updates
- RDF facilitates data re-use and re-purposing
- SPARQL raises the point of departure for an application
  - Expressive, open-ended query protocol
  - Support for unanticipated queries
- Benefits for the future
  - Linking to other data sources
  - Querying genes using the Fly Anatomy ontology
  - Magic of inference

# Costs & Risks

- Mapping data to RDF requires expertise and experience
- Expressive query protocol is a double-edged sword
- Performance is good for some queries, not for others…

# Performance

- Loading: Our datasets ~175 million triples
  - Jena / TDB gives much better load performance (~15-30K tps), on 64 bit system with Amazon EBS storage (~3hrs)
- Querying:
  - Good enough for real time user interaction, e.g., <1s for single gene search, 1-4s for multigene search (unions)
  - No significant slowdown when scale from 10m to 175m triples
- Text matching and case insensitive search
  - Problems with using SPARQL regex filter, the only mechanism for case-insensitive search in SPARQL
  - Pre-generated lower-case gene names and loaded into the FlyBase RDF DB
  - Tried with OpenLink Virtuoso, still ~10 seconds for a case-insensitive search

# Future directions

- Adding new data sources:
  - Gene expression data from EBI ArrayExpress
- More applications:
  - Find out all the gene expression images of its neighbours
  - Find out all the genes related to “blood pressure”
  - …
- Linked data (dereferencable, follow-your nose)
  - We're thinking about this, but our application does not currently need it
- How to control and predict quality of service for open SPARQL endpoints

# Acknowledgements

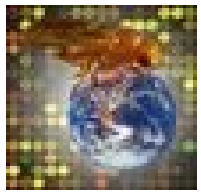

- Alistair Miles, Graham Klyne and David Shotton
- Dr Helen White-Cooper and her research group
- BBSRC for funding building the FlyTED database
- BDGP, FlyAtlas and FlyBase for making the data available
- JISC, for funding the FlyWeb project
- EPSRC, for funding the trip to DILS
- The Jena team, esp. Andy Seaborne


JISC

EPSRC

bbsrc

BDGP



FlyBase