BioMart

Data integration and retrieval made easy

Damian Smedley
European Bioinformatics Institute
Changing research focus

- The increase in high-throughput genomic technologies
- Growing sophistication of the user
- Research questions involving big datasets
  - Multi-species
  - Multi-experiments
  - Multi-datasets
- Data sources distributed
Solutions

- Bioinformatics support
  - Processing data files
  - Use third party software
  - In house processing of data

- No bioinformatics support?
- BioMart: one-stop shop for biological data
- For scientists with no programming experience and bioinformaticians
BioMart

- Fast and flexible integration system
- Query optimised database
- Interactive user-friendly interfaces (MartView, MartExplorer, MartShell)
- Allows user to group and refine biological data based upon many criteria
BioMart

- Tabulated data or FASTA sequence output in text, HTML or Excel formats
- First applied to Ensembl data (Ensembl, Vega and Est genes, SNPs): EnsMart
- BioMart includes UniProt proteomes and MSD protein structure data
- ArrayExpress soon
Talk summary

- BioMart interfaces: data access
- Usage examples
- System overview
Data access
Gene filters/attributes

- Region: chromosome position, band or marker
- External identifiers including microarray probes
- Gene Ontology and expression vocabulary terms
- Multi species: orthologs and upstream regions
- Protein and family identifiers
Gene filters/attributes

- Gene associated SNPs: location, synonymous status, ka_ks ratio
- Transcript sequences:
  - Coding
  - cDNA
  - Peptide
  - Exons
  - UTRs and upstream/downstream
  - User-specified flanking sequence
SNP filters and attributes

- Region
- Validation status
- Frequency data and population status
- Location in genes: coding, intronic etc
- SNP sequences
Usage examples
Candidate gene identification
SNPs for candidate genes

<table>
<thead>
<tr>
<th>SNP ID</th>
<th>Gene</th>
<th>Population</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENST00000163781.1</td>
<td>ENST00000163781.1</td>
<td>NORTH AMERICA BC_12_C:G 0.69</td>
<td>0.69</td>
</tr>
<tr>
<td>ENST00000163781.1</td>
<td>ENST00000163781.1</td>
<td>EAST ASIA BC_12_A:G 0.61</td>
<td>0.61</td>
</tr>
<tr>
<td>ENST00000163781.1</td>
<td>ENST00000163781.1</td>
<td>N XMTH ANE</td>
<td></td>
</tr>
</tbody>
</table>
Multi species
System Overview
BioMart

- Schema specification
- XML-based configuration
- Admin tools
  - Configuration/Building
- Data access
  - Libraries and interfaces (Perl, Java)
‘Reversed Star’ schema
Key features

- **Generic**
  - Universal BioMart data model
  - Query-based interface
  - No data dependent abstractions

- **Network scalability**
  - Query optimised schema

- **Platform portability**
  - Automatic, simple SQL
Key abstractions of generic system

Mart

Dataset

Attribute

Filter

GENE CENTRAL

gene_id(PK)
gene_stable_id
gene_start
gene_chrom_end
chromosome
gene_display_id
description
Deploying BioMart

Source databases → Transformation → Mart → Configuration

Specific scripts for each database
MartEditor

XML
Admin tools

- **MartEditor**
  - XML editor with build-in system logic
  - Configure existing interfaces
  - Automatically create new, ‘naive’ configuration
  - Handles database updating of XML for new releases
BioMart - a distributed architecture

MySQL  ORACLE  PostgreSQL

XML  XML  XML

XML  XML  XML

ANSI SQL

XML  XML  XML
MartShell (MQL)

- Uses Mart Query Language (MQL) to generate queries:

  using <dataset> get <attributes> where <filters>

- Can chain datasets together:

  using Dataset1 get Attribute1 where Filter1=var1 as q;
  using Dataset2 get Attribute2 where Filter2=var2 and filter3 in q

- Can script and pipe:

  martshell.sh -E MQLscript.mql > results.txt
  martshell.sh -E MQLscript.mql | wc
MartShell examples

MartShell> using MSD.msd get pdb_id where resolution_less < 1.5 and has_ec_info only;
  193l
  194l
  1arb ...

MartShell> using MSD.msd get pdb_id where resolution_less < 1.5 and has_ec_info only as q;
MartShell> using Ensembl.hsapiens_gene_ensembl get sequence transcript_flanks+1000 where pdb in q;

ENST00000270142.2  ENSG00000142168.2
  strand=forward  chr=21 assembly=NCBI34
downstream flanking sequence of transcript only
       AAACTAAATTAGCTCTGATACTTATTTATATAAACAGCTTCAGTGGAA
....
MartShell examples (cont)

MartShell> using Ensembl.hsapiens_gene_ensembl
get gene_stable_id, hugo, go_description
where chr_name = 3 and
    3.band_start = q22.1 and
    3.band_end = q22.3
and est.anatomical_site = retina;

ENSG00000051382  PIK3CB phosphoinositide 3-kinase complex
ENSG00000163914  RHOG-protein coupled photoreceptor activity ...
What do you get?

- Flexible interfaces configurable according to your spec
- ‘Performance-assured’ data retrieval
- Query chaining across data sources
- Administrator tools for modifying and deploying the system
BioMart - an open project

- All code and data freely available
  - Website
    - www.ebi.ac.uk/biomart
    - www.ebi.ac.uk/biomart/martview
  - Public MySQL server
    - martdb.ebi.ac.uk
  - Ftp
    - ftp.ebi.ac.uk
- Mailing lists
  - mart-dev
  - Mart-announce
Acknowledgements

- Arek Kasprzyk
- Darin London
- Damian Keefe
- Andreas Kahari
- Craig Melsopp
- Will Spooner
- Katerina Tzouvara