Database mining with biomaRt

Steffen Durinck

Lawrence Berkeley National Laboratory and
Division of Biostatistics, UC Berkeley

Statistics and Genomics seminar
August 2007

Overview

- The BioMart software suite
- biomaRt package for R
- ‘*Workshop style’* biomaRt example queries
  - Show the variety of different data types/questions that can be retrieved/answered

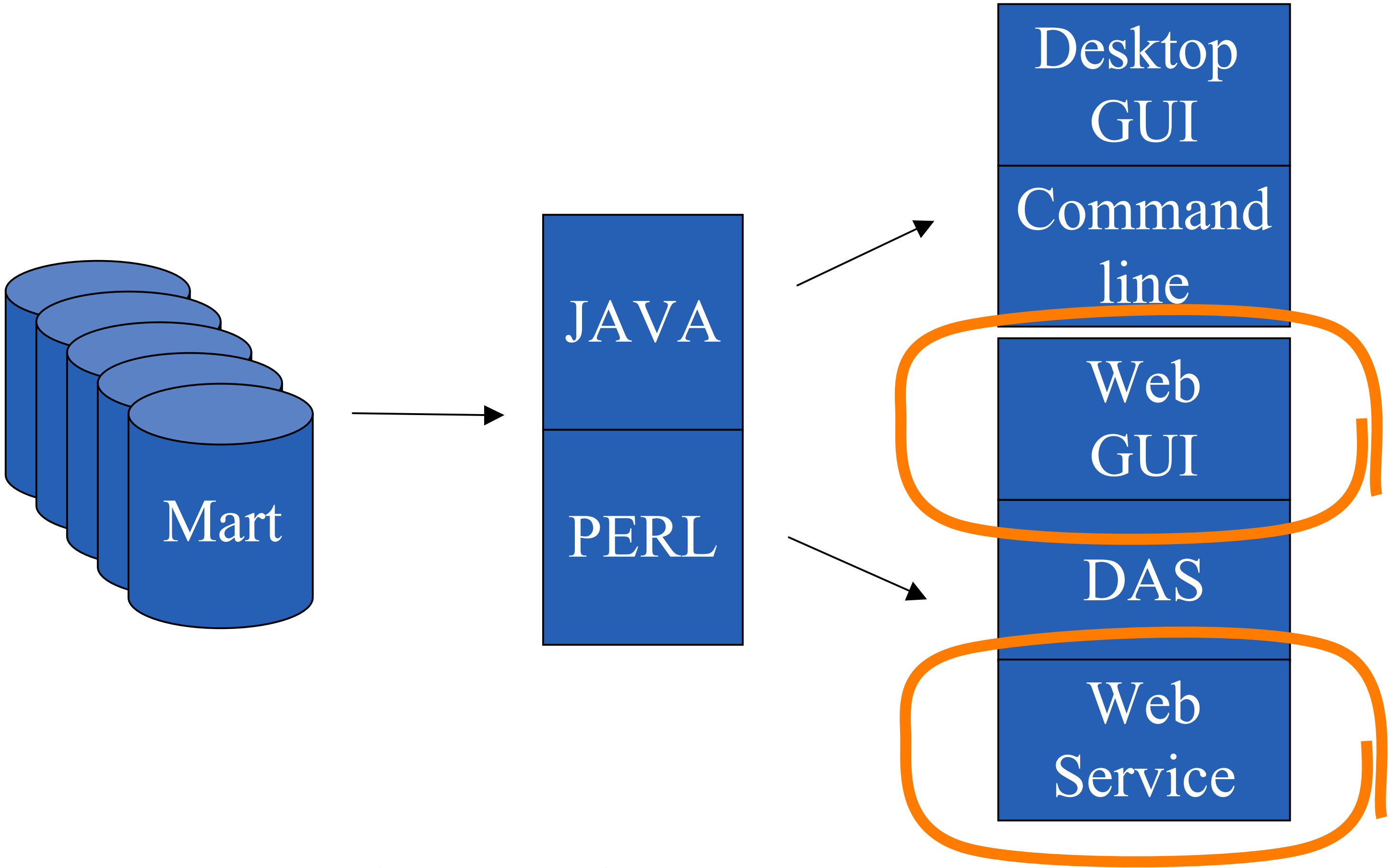
BioMart

- A query-oriented data management system
- Developed jointly by:
  - European Bioinformatics Institute (EBI)
  - Cold Spring Harbor Laboratory (CSHL)
- Originally developed for the Ensembl project but has now been generalized

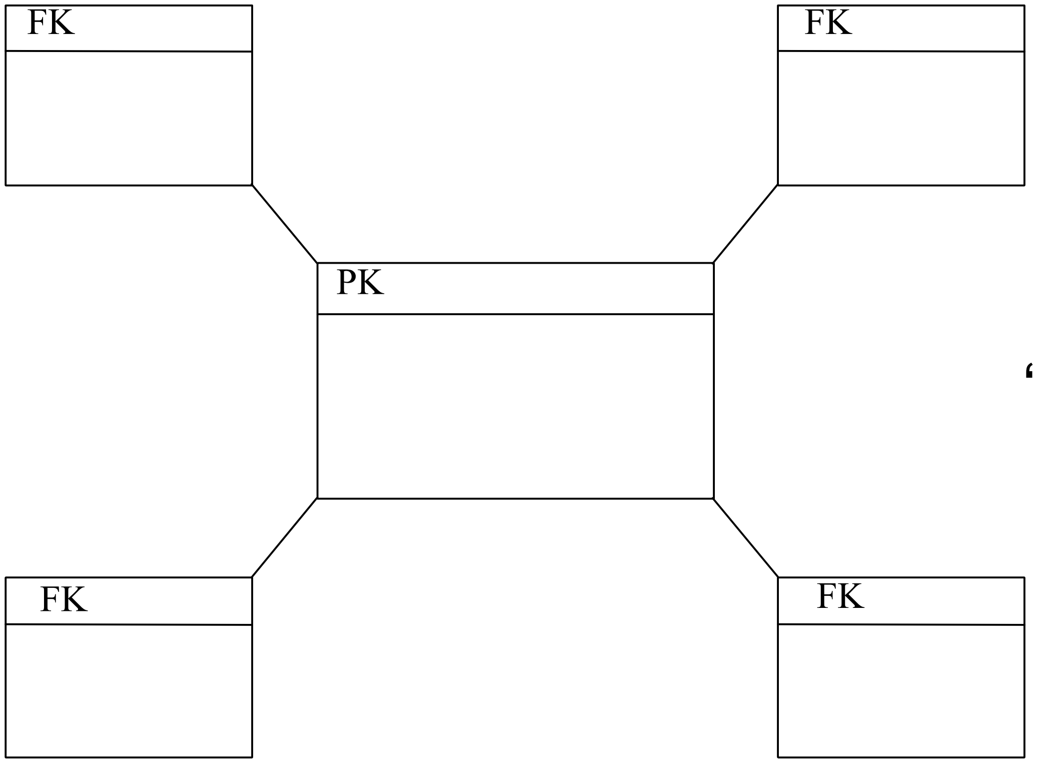
BioMart

- Aim:
  - To develop a generic data management system that works well for biology
    - ‘Data mining’
    - Query optimization
    - Lists
    - Data federation

BioMart Data Flow



BioMart Data Model



‘reverse star’

BioMart databases

- De-normalized
- Tables with ‘redundant’ information
- Query optimized
- Fast and flexible

- Well suited for batch querying

bio mart

BIOCONDUCTOR

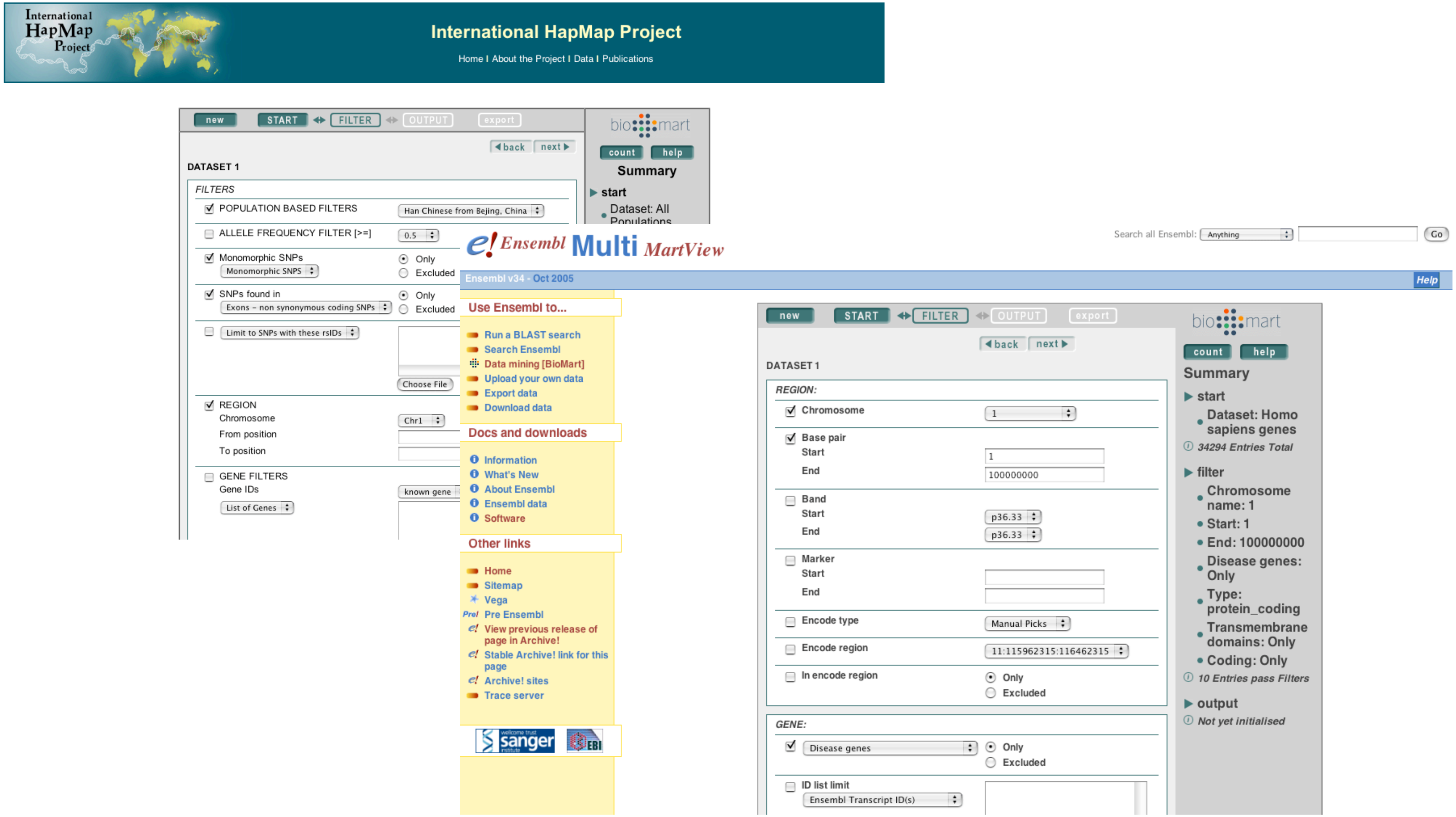
Examples of BioMart deployers

Statistics and Genomics seminar
August 2007

biomart

BIOCONDUCTOR

Genomic data


International HapMap Project
International HapMap Project
Home | About the Project | Data | Publications
new
START
FILTER
OUTPUT
export
back
next
biomart
count
help

DATASET 1

FILTERS

- ☑ POPULATION BASED FILTERS: Han Chinese from Bejing, China
- □ ALLELE FREQUENCY FILTER [>=]: 0.5
- ☑ Monomorphic SNPs: Monomorphic SNPS; ● Only ○ Excluded
- ☑ SNPs found in: Exons – non synonymous coding SNPs; ● Only ○ Excluded
- □ Limit to SNPs with these rsIDs
- ☑ REGION
  - Chromosome: Chr1
  - From position
  - To position
- □ GENE FILTERS
  - Gene IDs: known gene
  - List of Genes

Summary

- start
  - Dataset: All Populations

DATASET 1

REGION:

- ☑ Chromosome: 1
- ☑ Base pair
  - Start: 1
  - End: 100000000
- □ Band
  - Start: p36.33
  - End: p36.33
- □ Marker
  - Start
  - End
- □ Encode type: Manual Picks
- □ Encode region: 11:115962315:116462315
- □ In encode region: ● Only ○ Excluded

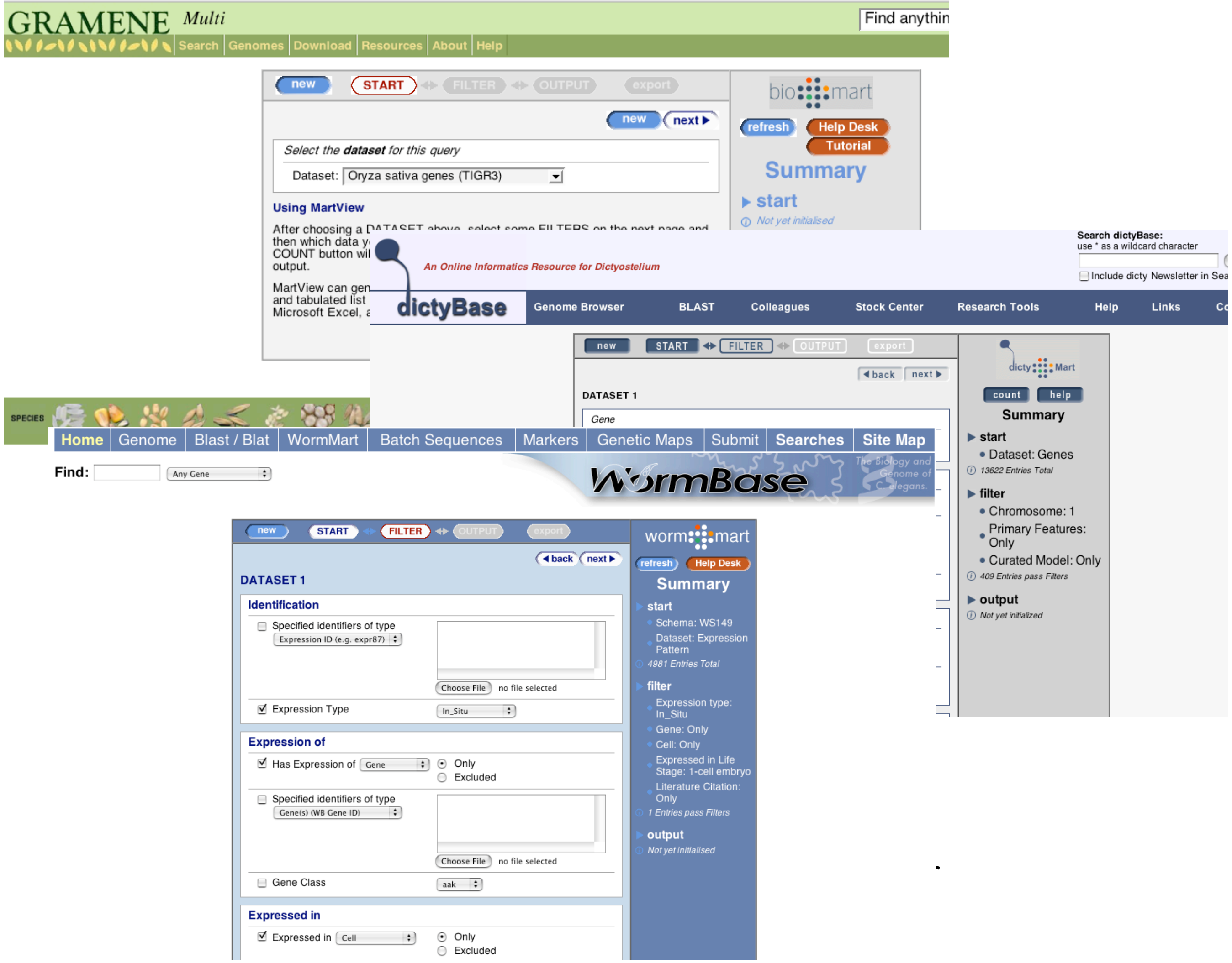
GENE:

- ☑ Disease genes: ● Only ○ Excluded
- □ ID list limit: Ensembl Transcript ID(s)

Summary

- start
  - Dataset: Homo sapiens genes
  - 34294 Entries Total
- filter
  - Chromosome name: 1
  - Start: 1
  - End: 100000000
  - Disease genes: Only
  - Type: protein_coding
  - Transmembrane domains: Only
  - Coding: Only
  - 10 Entries pass Filters
- output
  - Not yet initialised

Model organism databases



*Select the **dataset** for this query*

Dataset: Oryza sativa genes (TIGR3)

Using MartView

After choosing a DATASET above, select some FILTERS on the next page and then which data y COUNT button wil output.

MartView can gen and tabulated list Microsoft Excel, a

Summary

▶ start

Not yet initialised

An Online Informatics Resource for Dictyostelium

Search dictyBase:
use * as a wildcard character

□ Include dicty Newsletter in Sea

DATASET 1

Gene

Summary

▶ **start**
- Dataset: Genes

13622 Entries Total

▶ **filter**
- Chromosome: 1
- Primary Features: Only
- Curated Model: Only

409 Entries pass Filters

▶ **output**

Not yet initialized

Find: Any Gene

DATASET 1

Identification

□ Specified identifiers of type Expression ID (e.g. expr87)

no file selected

☑ Expression Type In_Situ

Expression of

☑ Has Expression of Gene ● Only ○ Excluded

□ Specified identifiers of type Gene(s) (WB Gene ID)

no file selected

□ Gene Class aak

Expressed in

☑ Expressed in Cell ● Only ○ Excluded

Summary

start
- Schema: WS149
- Dataset: Expression Pattern

4981 Entries Total

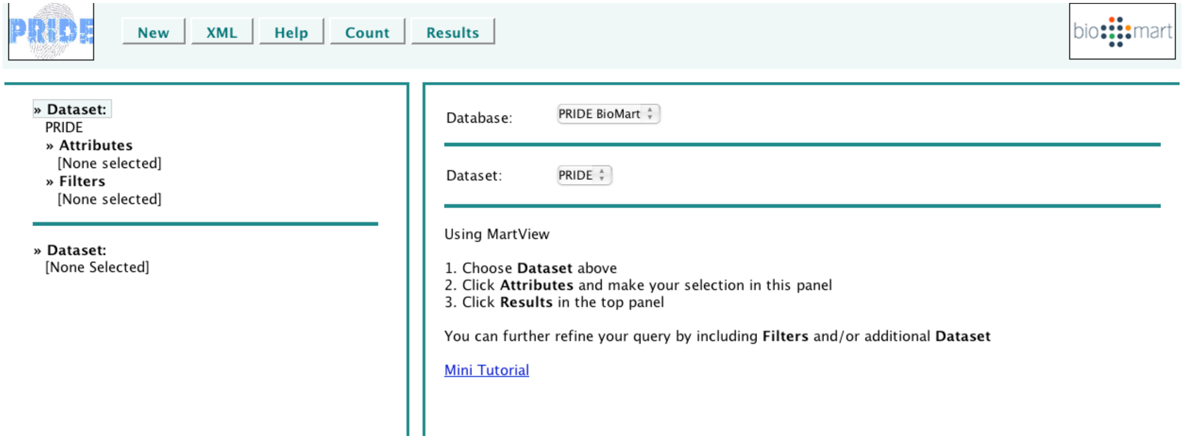
filter
- Expression type: In_Situ
- Gene: Only
- Cell: Only
- Expressed in Life Stage: 1-cell embryo
- Literature Citation: Only

1 Entries pass Filters

output

Not yet initialised

Proteomics



» Dataset:
PRIDE
» Attributes
[None selected]
» Filters
[None selected]

» Dataset:
[None Selected]

Database: PRIDE BioMart

Dataset: PRIDE

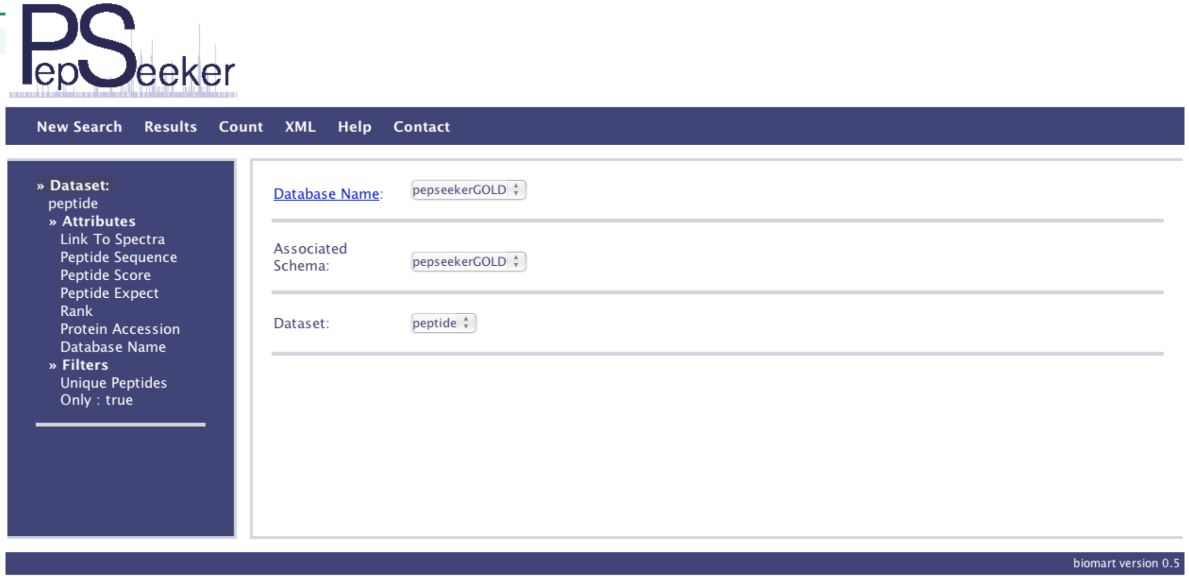
Using MartView

1. Choose **Dataset** above
2. Click **Attributes** and make your selection in this panel
3. Click **Results** in the top panel

You can further refine your query by including **Filters** and/or additional **Dataset**

Mini Tutorial

biomart version 0.5



» Dataset:
peptide
» Attributes
Link To Spectra
Peptide Sequence
Peptide Score
Peptide Expect
Rank
Protein Accession
Database Name
» Filters
Unique Peptides Only : true

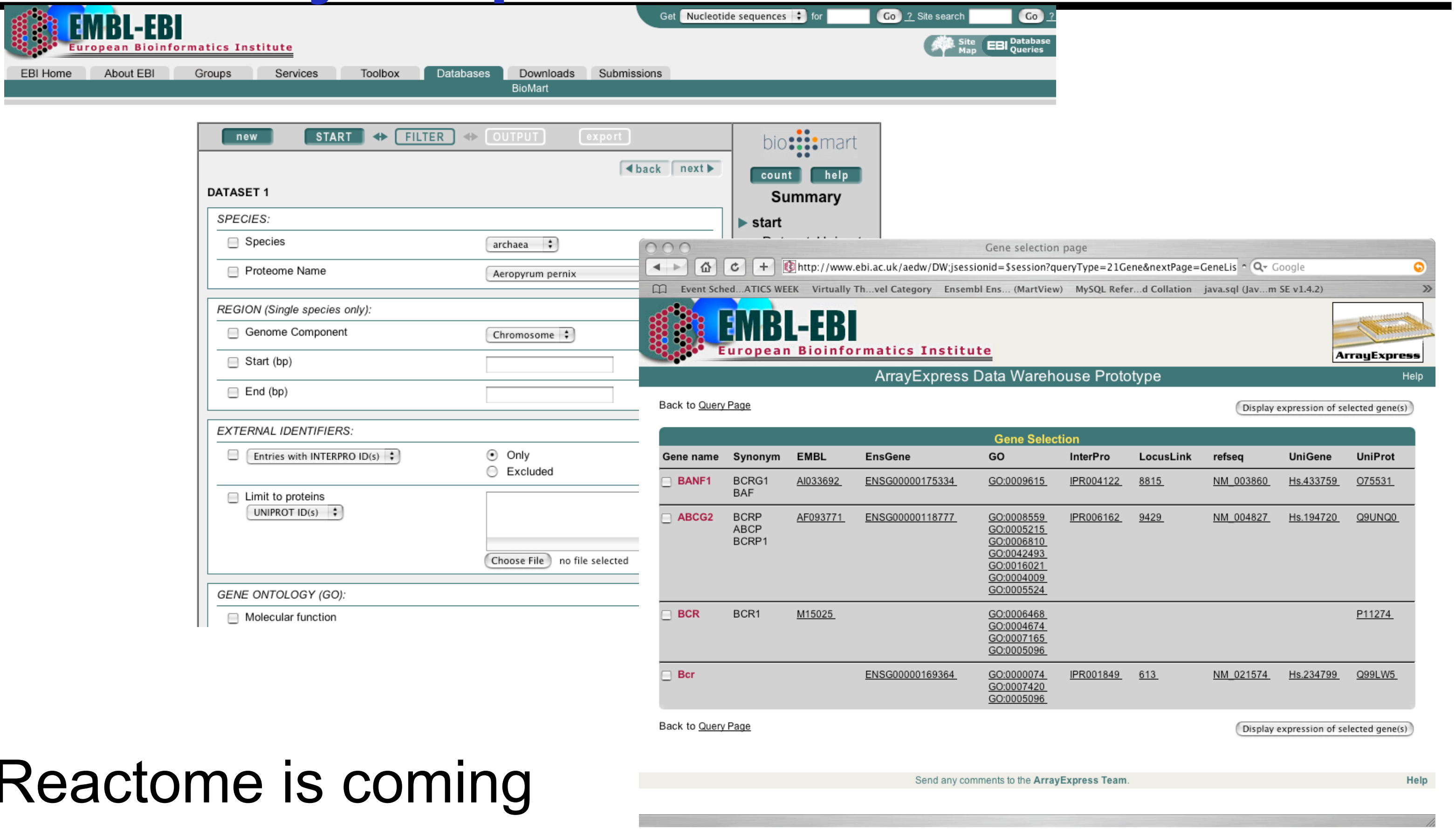
Database Name: pepseekerGOLD

Associated Schema: pepseekerGOLD

Dataset: peptide

biomart version 0.5

Uniprot, ArrayExpress,Reactome


EMBL-EBI
European Bioinformatics Institute
Get Nucleotide sequences for Go ? Site search Go ?
Site Map
EBI Database Queries
EBI Home
About EBI
Groups
Services
Toolbox
Databases
Downloads
Submissions
BioMart
new
START
FILTER
OUTPUT
export
back
next
count
help
Summary
start
Gene selection page
http://www.ebi.ac.uk/aedw/DW;jsessionid=$session?queryType=21Gene&nextPage=GeneLis
Google
Event Sched...ATICS WEEK
Virtually Th...vel Category
Ensembl Ens... (MartView)
MySQL Refer...d Collation
java.sql (Jav...m SE v1.4.2)
ArrayExpress
Help
Choose File
Display expression of selected gene(s)

DATASET 1

SPECIES:

- □ Species: archaea
- □ Proteome Name: Aeropyrum pernix

REGION (Single species only):

- □ Genome Component: Chromosome
- □ Start (bp)
- □ End (bp)

EXTERNAL IDENTIFIERS:

- □ Entries with INTERPRO ID(s) — ● Only ○ Excluded
- □ Limit to proteins: UNIPROT ID(s) — no file selected

GENE ONTOLOGY (GO):

- □ Molecular function

ArrayExpress Data Warehouse Prototype

Back to Query Page

Gene Selection

Gene name	Synonym	EMBL	EnsGene	GO	InterPro	LocusLink	refseq	UniGene	UniProt
□ BANF1	BCRG1 BAF	AI033692	ENSG00000175334	GO:0009615	IPR004122	8815	NM_003860	Hs.433759	O75531
□ ABCG2	BCRP ABCP BCRP1	AF093771	ENSG00000118777	GO:0008559 GO:0005215 GO:0006810 GO:0042493 GO:0016021 GO:0004009 GO:0005524	IPR006162	9429	NM_004827	Hs.194720	Q9UNQ0
□ BCR	BCR1	M15025		GO:0006468 GO:0004674 GO:0007165 GO:0005096					P11274
□ Bcr			ENSG00000169364	GO:0000074 GO:0007420 GO:0005096	IPR001849	613	NM_021574	Hs.234799	Q99LW5

Back to Query Page

Send any comments to the ArrayExpress Team.

Reactome is coming

biomaRt package for R

Statistics and Genomics seminar
August 2007

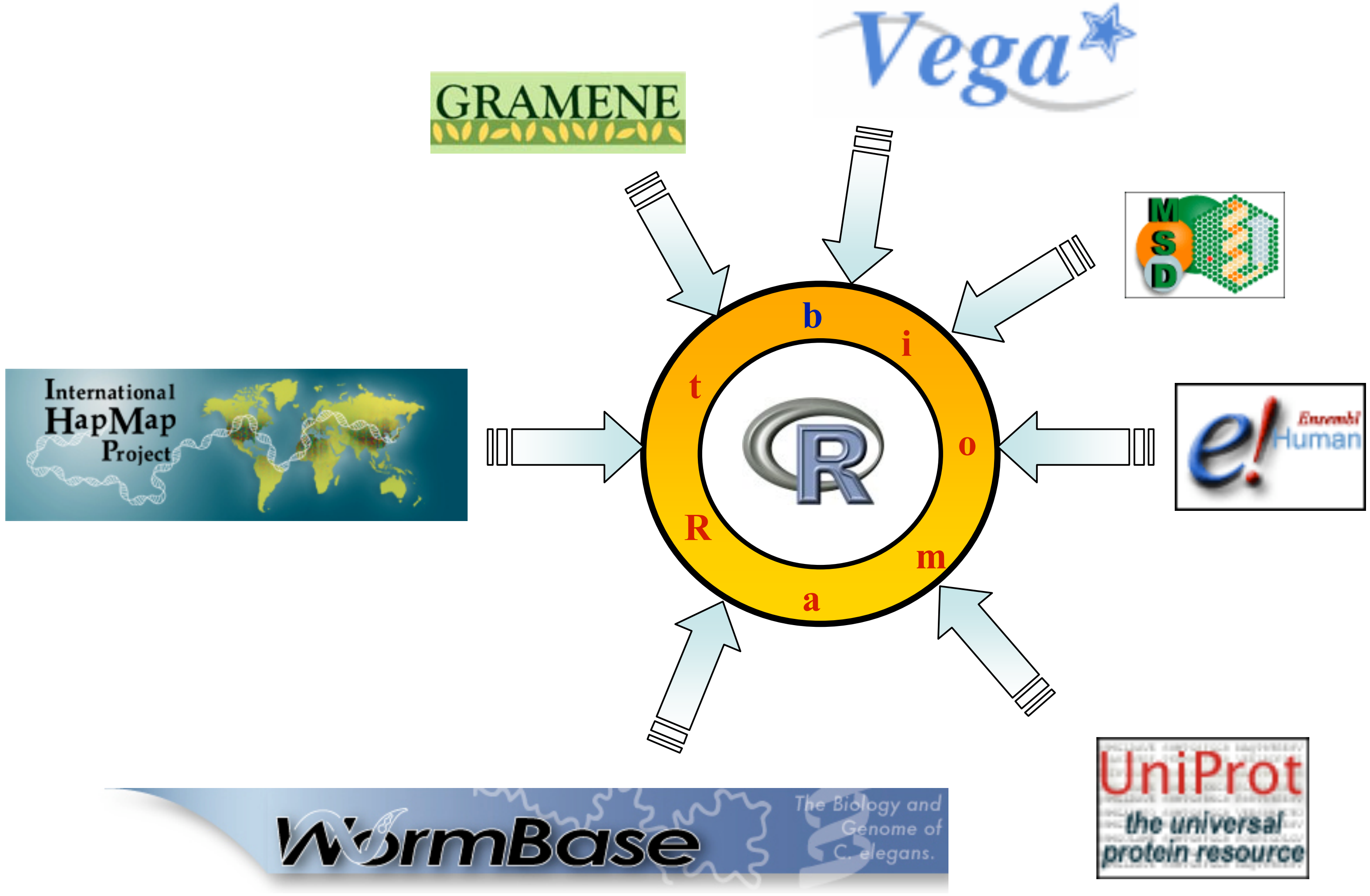
biomaRt

- R interface to BioMart databases
- Performs online queries
- Current release version 1.10.1
- Depends on Rcurl and XML packages
- Optional RMySQL

bio mart

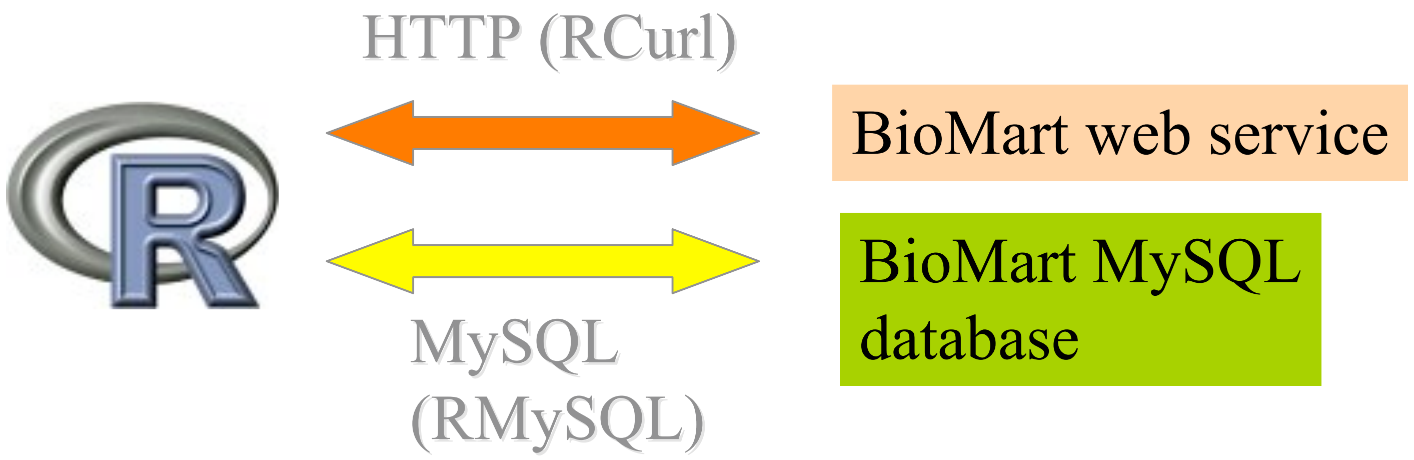
BIOCONDUCTOR

biomaRt - aim



biomaRt - db access

- Direct HTTP queries to BioMart web services
- MySQL queries to BioMart databases



HTTP (RCurl)

BioMart web service

BioMart MySQL database

MySQL (RMySQL)

Installing biomaRt

- Platforms on which biomaRt has been installed:
  - Linux
  - OSX
  - Windows

Installing biomaRt

```
> source( "http://www.bioconductor.org/biocLite.R")

> biocLite("biomaRt")

Running biocinstall version 2.0.8 with R version 2.5.0
Your version of R requires version 2.0 of Bioconductor.
also installing the dependencies 'XML', 'RCurl'
```

List available BioMart databases

```
> library(biomaRt)
Loading required package: XML
Loading required package: Rcurl
> listMarts()
```

List available BioMarts

	name	version
1	ensembl	ENSEMBL 46 GENES (SANGER)
2	compara_mart_homology_46	ENSEMBL 46 HOMOLOGY (SANGER)
3	compara_mart_pairwise_ga_46	ENSEMBL 46 PAIRWISE ALIGNMENTS (SANGER)
4	compara_mart_multiple_ga_46	ENSEMBL 46 MULTIPLE ALIGNMENTS (SANGER)
5	snp	ENSEMBL 46 VARIATION (SANGER)
6	genomic_features	ENSEMBL 46 GENOMIC FEATURES (SANGER)
7	vega	VEGA 21 (SANGER)
8	uniprot	UNIPROT PROTOTYPE (EBI)
9	msd	MSD PROTOTYPE (EBI)
10	ENSEMBL_MART_ENSEMBL	GRAMENE (CSHL)
11	wormbase176	WORMBASE (CSHL)
12	dicty	DICTYBASE (NORTHWESTERN)
13	rgd_mart	RGD GENES (MCW)
14	SSLP_mart	RGD MICROSATELLITE MARKERS (MCW)
15	pepseekerGOLD_mart	PEPSEEKER (UNIVERSITY OF MANCHESTER)
16	pride	PRIDE (EBI)
17	Pancreatic_Expression	PANCREATIC EXPRESSION DATABASE (INST OF CANCER)

Ensembl

- Ensembl is a joint project between EMBL - European Bioinformatics Institute (EBI) and the Wellcome Trust Sanger Institute (WTSI)
- A software system which produces and maintains automatic annotation on selected eukaryotic genomes.
- http://www.ensembl.org

Ensembl - Datasets

> ensembl=useMart("ensembl")

BioMarts usually contain different datasets

In Ensembl every species is a different dataset

Ensembl - Datasets

```
> listDatasets(ensembl)
```

```
           dataset                               description                         version
1    oanatinus_gene_ensembl         Ornithorhynchus anatinus genes (OANA5)         OANA5
2    gaculeatus_gene_ensembl         Gasterosteus aculeatus genes (BROADS1)  ROADS1
3    cporcellus_gene_ensembl         Cavia porcellus genes (GUINEAPIG)            GUINEAPIG
4    lafricana_gene_ensembl          Loxodonta africana genes (BROADE1)          BROADE1
5  stridecemlineatus_gene_ensembl Spermophilus tridecemlineatus genes (SQUIRREL)      SQUIRREL
6    scerevisiae_gene_ensembl        Saccharomyces cerevisiae genes (SGD1.01) SGD1.01
7    eeuropaeus_gene_ensembl         Erinaceus europaeus genes (HEDGEHOG)   HEDGEHOG
8    etelfairi_gene_ensembl          Echinops telfairi genes (TENREC)             TENREC
9    ptroglodytes_gene_ensembl       Pan troglodytes genes (CHIMP2.1)             CHIMP2.1
10  cintestinalis_gene_ensembl       Ciona intestinalis genes (JGI2)              JGI2
11  ocuniculus_gene_ensembl          Oryctolagus cuniculus genes (RABBIT)         RABBIT
12  hsapiens_gene_ensembl            Homo sapiens genes (NCBI36)                  NCBI36
13  ggallus_gene_ensembl             Gallus gallus genes (WASHUC2)                WASHUC2

              …                                …                                …
```

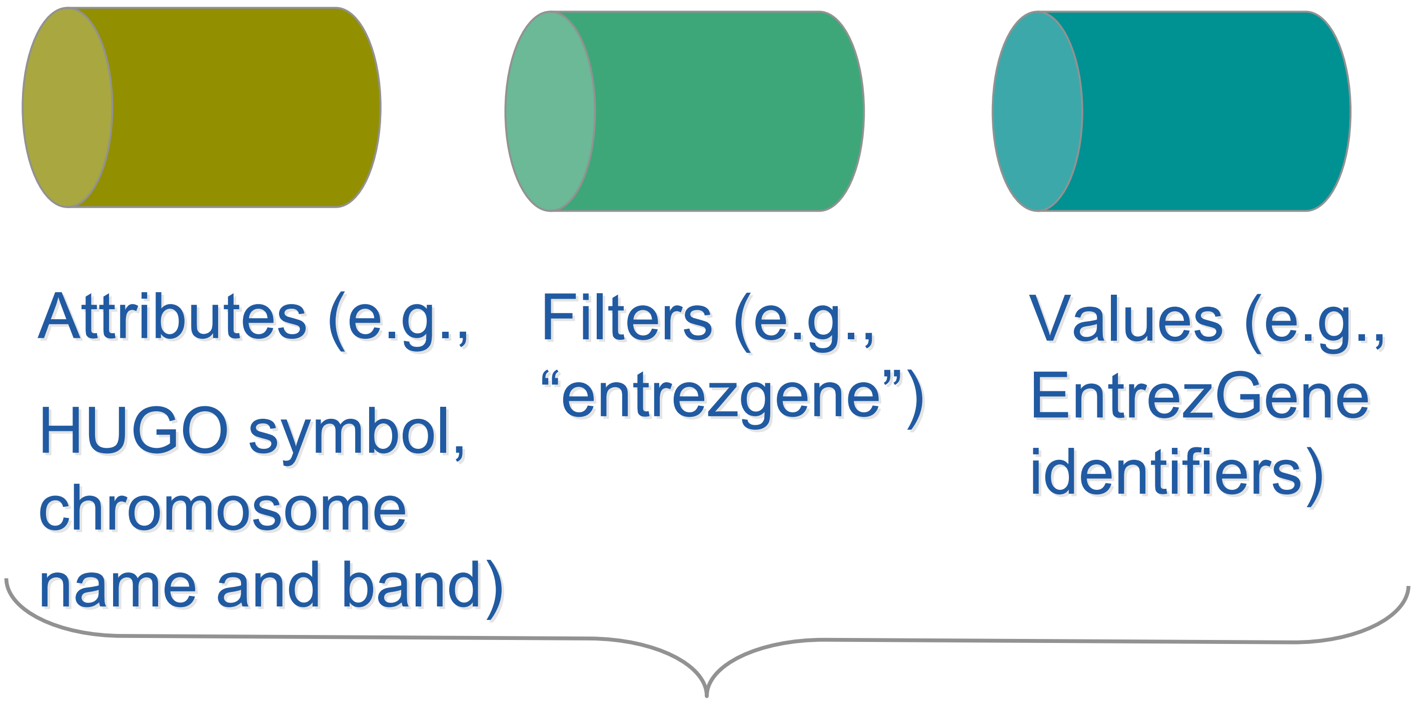
Ensembl - Datasets

A dataset can be selected using the useMart function

```
> ensembl = useMart(“ensembl”,
   dataset=“hsapiens_gene_ensembl)
```

```
Checking attributes and filters ... ok
```

biomaRt query



Attributes (e.g., HUGO symbol, chromosome name and band)

Filters (e.g., “entrezgene”)

Values (e.g., EntrezGene identifiers)

Three main biomaRt functions

- *listFilters*
  - – Lists the available filters
- *listAttributes*
  - – Lists the available attributes
- *getBM*
  - – Performs the actual query and returns a *data.frame*

listAttributes

```
> listAttributes(ensembl)
                    name                  description
1             affy_hcg110                  AFFY HCG110
2           affy_hg_focus                AFFY HG FOCUS
3      affy_hg_u133_plus_2           AFFY HG U133-PLUS-2
4           affy_hg_u133a                AFFY HG U133A
5        affy_hg_u133a_v2              AFFY HG U133Av2
6           affy_hg_u133b                AFFY HG U133B
7            affy_hg_u95a                 AFFY HG U95A
8          affy_hg_u95av2               AFFY HG U95AV2
9            affy_hg_u95b                 AFFY HG U95B
10           affy_hg_u95c                 AFFY HG U95C
11           affy_hg_u95d                 AFFY HG U95D
12           affy_hg_u95e                 AFFY HG U95E
13          affy_hugenefl               AFFY HUGENEFL
14          affy_u133_x3p                AFFY U133 X3P
15            agilent_cgh                  Agilent cgh
16          agilent_probe                Agilent Probe
17                   band                         Band
18                biotype                      Biotype
19                   ccds                      CCDS ID
```

Example 1 - Ensembl

- Annotate the following Affymetrix probe identifiers:

  211550_at, 202431_s_at, 206044_s_at

- From the human u133plus2 platform

- With HUGO gene nomenclature symbol (hgnc_symbol) and chromosomal location information

Example 1 - Ensembl

- Filters: *affy_hg_u133_plus_2*
- Attributes:

  affy_hg_u133_plus_2, chromosome_name, start_position, end_position, band, strand

- Values:

  211550_at, 202431_s_at, 206044_s_at

Example 1 - Ensembl

```
> affyids =
   c("211550_at","202431_s_at","206044_s_at")
> annotation =
   getBM(attributes=c("affy_hg_u133_plus_2","ense
   mbl_transcript_id","ensembl_gene_id","hgnc_sym
   bol","chromosome_name","start_position","end_p
   osition","band","strand"),
   filters="affy_hg_u133_plus_2", values=affyids,
   mart = ensembl)
```

Example 1 - Ensembl

```
>annotation

  affy_hg_u133_plus_2 ensembl_transcript_id ensembl_gene_id hgnc_symbol
1        211550_at       ENST00000344576   ENSG00000146648       EGFR
2       202431_s_at      ENST00000377970   ENSG00000136997       MYC
3       202431_s_at      ENST00000259523   ENSG00000136997       MYC
4       206044_s_at      ENST00000288602   ENSG00000157764       BRAF

  chromosome_name start_position end_position      band       strand
1            7        55054219          55242524      p11.2      1
2            8        128817498         128822853     q24.21      1
3            8        128817498         128822853     q24.21       1
4            7        140080754         140271033     q34        -1
```

Microarray probes & Ensembl

- Ensembl does an independent mapping of e.g. Affymetrix probe sequences to genomes
- If there is no clear match then that probe is not assigned to a gene
- If there are multple matches, all matches are returned

Using more than one filter

- getBM can be used with more than one filter
- Filters should be given as a vector
- Values should be a list of vectors where the position of each vector corresponds with the position of the associated filter in the filters argument

Example 2 - Ensembl

Retrieve all genes that are known to be involved in Diabetes Mellitus Type I or Type II and have transcription factor activity

Example 2 - Ensembl

1. Diabetes Mellitus type I MIM accession: *222100*
2. Diabetes Mellitus type II MIM accession: *125853*
3. GO id for “transcription factor activity”: *GO:0003700*

Example 2 - Ensembl

```
>diab=getBM(attributes=c("ensembl_gene_id","hgnc_symbol"),
       filters=c("mim_morbid_ac","go"),
       values=list(c("125853","222100"),"GO:0003700"),
       mart=ensembl)
> unique(diab)
   ensembl_gene_id  hgnc_symbol
1 ENSG00000139515         PDX1
3 ENSG00000108753         TCF2
4 ENSG00000135100         TCF1
```

Boolean filters

- Filters can be either numeric, string or boolean
- Boolean filters should have either TRUE or FALSE as values
  - TRUE: return all information that comply with the given filter (e.g. return only genes that have a hgnc_symbol)
  - FALSE: return all information that doesn't comply with the given filter (e.g. with no hgnc_symbol)

Example 3 - Ensembl

Retrieve all miRNAs known on chromosome 13 and their chromosomal locations

Example 3 - Ensembl

```
> miRNA =
    getBM(c("mirbase","ensembl_gene_id","start_position",
    "chromosome_name"),
    filters=c("chromosome_name","with_mirbase"),
    values=list(13,TRUE), mart=ensembl)
> miRNA
```

Example 3 - Ensembl

```
      mirbase         ensembl_gene_id    start_position chromosome_name
1  MI0000074    ENSG00000207560       90801447              13
2  MI0003637    ENSG00000207719       98806386              13
3  MI0000070    ENSG00000208006       49521110              13
4  MI0000076    ENSG00000199149       90801320              13
5  MI0003636    ENSG00000207858       89681437              13
6  MI0000073    ENSG00000207610       90801146              13
7  MI0000069    ENSG00000207718       49521256              13
8  MI0003635    ENSG00000207652       40282902              13
9  MI0000071    ENSG00000207745       90800860              13
10 MI0000072    ENSG00000199180       90800998              13
11 MI0000093    ENSG00000207968       90801569              13
```

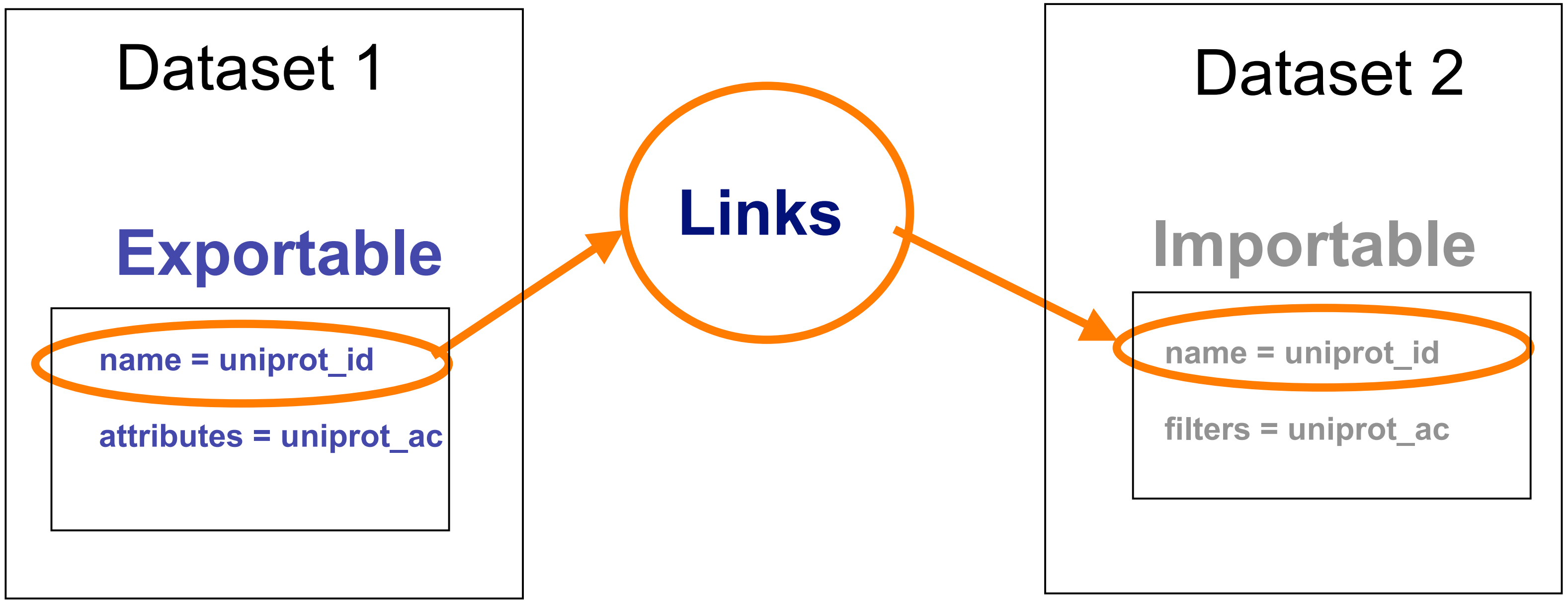
Example 4 - Ensembl

Retrieve all entrezgene identifiers on chromosome 22 that have a coding SNP

Example 4 - Ensembl

```
> entrez =
   getBM("entrezgene",filters=c("chromosome_name
   ” ,"with_coding_snp"),
   values=list(22,TRUE),mart=ensembl)
> entrez[1:5,]
>  entrez[1:5,]
[1] 649486  81061 440153 150160 150165
```

BioMart Data federation



Dataset 1

Exportable

name = uniprot_id

attributes = uniprot_ac

Links

Dataset 2

Importable

name = uniprot_id

filters = uniprot_ac

Homology - Ensembl

- The different species in Ensembl are interlinked
- biomaRt takes advantage of this to provide homology mappings between different species

Linking two datasets

- Two datasets (e.g. two species in Ensembl) can be linked to each other by using the *getLDS* (get linked dataset) function
- One has to connect to two different datasets and specify the linked dataset using *martL*, *filtersL*, *attributesL*, *valuesL* arguments

Example 5 - Ensembl

Given:

Human affy_hg_u95av2 platform:
976_s_at, 1888_s_at

? Retrieve human gene symbol and chicken chip Affymetrix identifiers of their homologs in chicken

Example 5 - Ensembl

```
> human=useMart("ensembl", dataset="hsapiens_gene_ensembl")
  Checking attributes and filters ... ok
> chicken=useMart("ensembl", dataset="ggallus_gene_ensembl")
  Checking attributes and filters ... ok
>getLDS(attributes=c("affy_hg_u95av2","hgnc_symbol"),
    filters="affy_hg_u95av2",
    values=c("1888_s_at","976_s_at"),mart=human,
    attributesL="affy_chicken", martL=chicken)

       V1              V2          V3
1  976_s_at         MAPK1       Gga.2163.1.S1_at
2  976_s_at         MAPK1       Gga.18672.1.S1_at
3 1888_s_at         KIT         Gga.606.1.S1_at
```

Example 6 - Ensembl

Select all genes (human gene symbols and mouse Ensembl gene identifiers) located on human chromosome 1 that are located on mouse chromosome 2

Example 6 - Ensembl

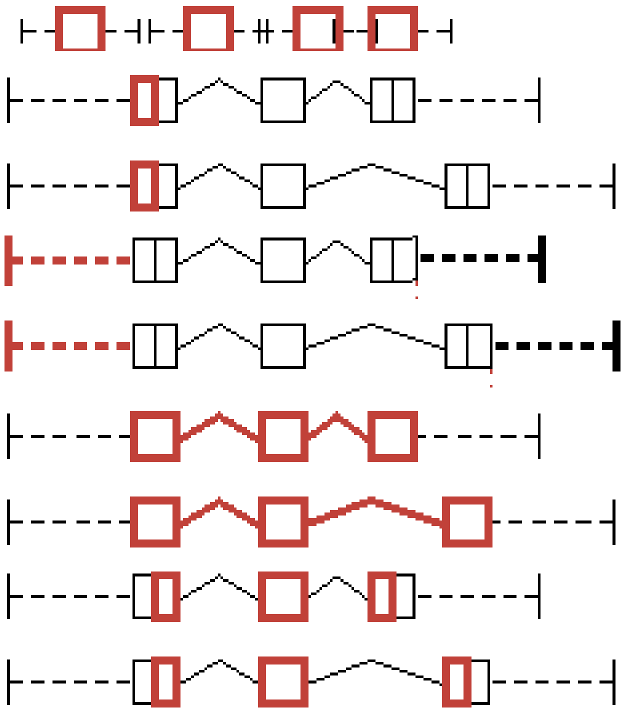
```
> mouse=useMart("ensembl", dataset="mmusculus_gene_ensembl")
  Checking attributes and filters ... ok
>out=getLDS(attributes=c("hgnc_symbol","ensembl_gene_id",
   "chromosome_name"), filters="chromosome_name",values=1,
   mart=human,
   attributesL=c("ensembl_gene_id","chromosome_name"),
   filtersL="chromosome_name", valuesL=2, martL=mouse )
> unique(out[1:10,])
       V1             V2            V3            V4           V5
1  SLC39A1    ENSG00000143570  1 ENSMUSG00000058850  2
5    VPS45    ENSG00000136631  1 ENSMUSG00000075362  2
7 PRAMEF19    ENSG00000204480  1 ENSMUSG00000025839  2
8 PRAMEF19    ENSG00000204480  1 ENSMUSG00000025838  2
```

getSequence

- Retrieving sequences from Ensembl can be done using the *getBM* function or the *getSequence* wrapper function
- Output of *getSequence* can be exported to FASTA file using the *exportFASTA* function

getSequence

- Available sequences in Ensembl:
  - Exon
  - 3'UTR
  - 5'UTR
  - Upstream sequences
  - Downstream sequences
  - Unspliced transcript/gene
  - Coding sequence
  - Protein sequence



getSequence

- Arguments of getSequence:
  - *id*: identifier
  - *type*: type of identifier used e.g. hgnc_symbol or affy_hg_u133_plus_2
  - *seqType*: sequence type that needs to be retrieved e.g. gene_exon, coding, 3utr, 5utr,
  - *upstream/downstream*: specify number of base pairs upstream/downstream that need to be retrieved

Example 7 - Ensembl

Retrieve all exons of CDH1

Example 7 - Ensembl

```
 > seq = getSequence(id="CDH1",
    type="hgnc_symbol",seqType="gene_exon", mart = ensembl)
> seq[1,]

    gene_exon
 1
    TACAAGGGTCAGGTGCCTGAGAACGAGGCTAACGTCGTAATCAC
    CACACTGAAAGTGACTGATGCTGATGCCCCCAATACCCCAGCGT
    GGGAGGCTGTATACACCATATTGAATGATGATGGTGGACAATTTG
    TCGTCACCACAAATCCAGTGAACAACGATGGCATTTTGAAAACAG
    CAAAG
  hgnc_symbol
1       CDH1
```

Example 8 - Ensembl

Retrieve 2000bp sequence upstream of the APC and CUL1 translation start site and count number of E-box motifs to verify possible regulation by MYC transcription factor

E-box motif: 5’-CACGTG-3’

Example 8 - Ensembl

```
>promoter=getSequence(id=c("APC","CUL1"),type=
   "hgnc_symbol",
   seqType="coding_gene_flank",upstream =2000,
   mart=ensembl)
> ebox =
   strsplit(as.character(promoter[1,]),"CACGTG")
> length(ebox)-1
[1] 1
```

SNP BioMart

- dbSNP mapped to Ensembl

```
> snp = useMart(“snp”, dataset=“hsapiens_snp”))
```

Example 9 - SNP

Retrieve all refsnp_ids and their alleles and positions that are located on chromosome 8 and between bp 148350 and 148612.

Example 9 - SNP

```
>out=getBM(attributes=c("refsnp_id","allele","chrom_start"),
   filters=c("chr_name","chrom_start","chrom_end"),
   values=list(8,148350,148612), mart=snp)
> out[1:5,]
   refsnp_id      allele      chrom_start
1  rs1134195       G/T         48394
2  rs4046274       C/A        148394
3  rs4046275       A/G        148411
4  rs13291        C/T         148462
5  rs1134192      G/A         148462
```

UniProt

the universal protein resource

- UniProt (Universal Protein Resource) is the world's most comprehensive catalog of information on proteins.
- It is a central repository of protein sequence and function created by joining the information contained in Swiss-Prot, TrEMBL, and PIR.

Example 10 - Uniprot

Which is the longest annotated protein in human?

Example 10 - Uniprot

```
>unip = useMart(“uniprot”,dataset=“uniprot”)
> lengths = getBM(c("protein_name","length"),
   filters=c("proteome_name","length_greater"),
   values=list("Homo sapiens",4000),mart=unip)
> longest = which(lengths[,2] == max(lengths[,2]))
> lengths[longest,]
     protein_name length
4832        Titin     34350
```

Example 11 - Uniprot

Retrieve proteins that have an alpha-helix and have a length smaller than 100 AA

Example 11 - Uniprot

```
>unip = useMart("uniprot", dataset="uniprot")
> proteins = getBM(c("protein_name","length"),
   filters=c("has_helix","length_smaller"),
   values=list(TRUE,100), mart=unip)
> proteins[1:5,]
                   protein_name                    length
1 Transition state regulatory protein abrB          96
2               Acyl carrier protein                77
3               HPr-like protein crh                85
4             Cold shock protein cspB               67
5            Germination protein gerE               74
```

Example 12 - Uniprot

Determine the INTERPRO protein domains of PDGFRA

Example 12 - Uniprot

```
> interpro = getBM("short_name",filters="gene_name",
   values="PDGFRA", mart=unip)
> unique(interpro[,1])
 [1] "Prot_kinase"      "Tyr_pkinase"      "RecepttyrkinsIII"
 [4] "Ig_c2"            "Ig-like"          "Tyr_pkinase_AS"
 [7] "VEGFR"            "Kinase_like"      "Ser_thr_pkinase"
[10] "Ig"
```

Gramene

- Gramene is a curated, open-source, data resource for comparative genome analysis in the grasses.
- Rice, Maize and Arabidopsis

Example 13 - Gramene

Retrieve affy ATH1 ids and CATMA ids that map to the *Arabidopsis thaliana* chromosome 1 between basepair 30.000 and 41.000

Example 13 - Gramene

```
>gramene =
  useMart("ENSEMBL_MART_ENSEMBL",
  dataset="athaliana_gene_ensembl")
>getBM(c("affy_ath1_id","catma_tigr5_id"),
  filters=c("chromosome_name","start","end")
  , values=list("1", "30000","41000"),
  mart=gramene)
```

Example 13 - Gramene

```
  affy_ath1_id catma_tigr5_id
1    261579_at   CATMA1a00040
2    261569_at   CATMA1a00045
3    261569_at   CATMA1a00045
4    261569_at   CATMA1a00045
5    261576_at   CATMA1a00050
6    261576_at   CATMA1a00050
```

Wormbase

- Database on the genetics of C elegans and related nematodes.

Example 14 - Wormbase

Determine the RNAi constructs and the observed phenotypes for the gene unc-26 with wormbase gene id: WBGene00006763

Example 14 - Wormbase

```
> worm = useMart("wormbase176",
                          dataset="wormbase_rnai")


> pheno =
   getBM(c(“rnai”,”phenotype_primary_name”),
   filters=“gene”, values=“WBGene00006763”,
   mart=worm)
```

Example 14 - Wormbase

```
>pheno
  rnai              phenotype_primary_name
1  WBRNAi00021278                  slow_growth
2  WBRNAi00021278 postembryonic_development_abnormal
3  WBRNAi00021278              embryonic_lethal
4  WBRNAi00021278                 larval_lethal
5  WBRNAi00021278                 larval_arrest
6  WBRNAi00021278              maternal_sterile
7  WBRNAi00021278                     Abnormal
8  WBRNAi00021278               sterile_progeny
9  WBRNAi00026915                  slow_growth
10 WBRNAi00026915 postembryonic_development_abnormal
```

Locally installed BioMarts

- Main use case currently is to use biomaRt to query public BioMart servers over the internet
- But you can also install BioMart server locally, populated with a copy of a public dataset (particular version), or populated with your own data
- Versioning is supported by naming convention

Discussion

- Using biomaRt to query public web services gets you started quickly, is easy and gives you access to a large body of metadata in a uniform way
- Need to be online
- Online metadata can change behind your back; although there is possibility of connecting to a particular, immutable version of a dataset

Acknowledgements

- EBI
  - Wolfgang Huber
  - Arek Kasprzyk
  - Ewan Birney
  - Alvis Brazma
  - BioMart team

- Bioconductor users