Online queries to BioMart web services through biomaRt

Steffen Durinck[1], Wolfgang Huber[2]

1. NCI/NIH, Gaithersburg, Maryland, USA

2. EBI, Hinxton-Cambridge, UK

This workshop

- What is BioMart?
- Example BioMart databases
- Overview of the biomaRt package
  - Simple biomaRt functions
  - Generic biomaRt functions

Hands - on

BioMart

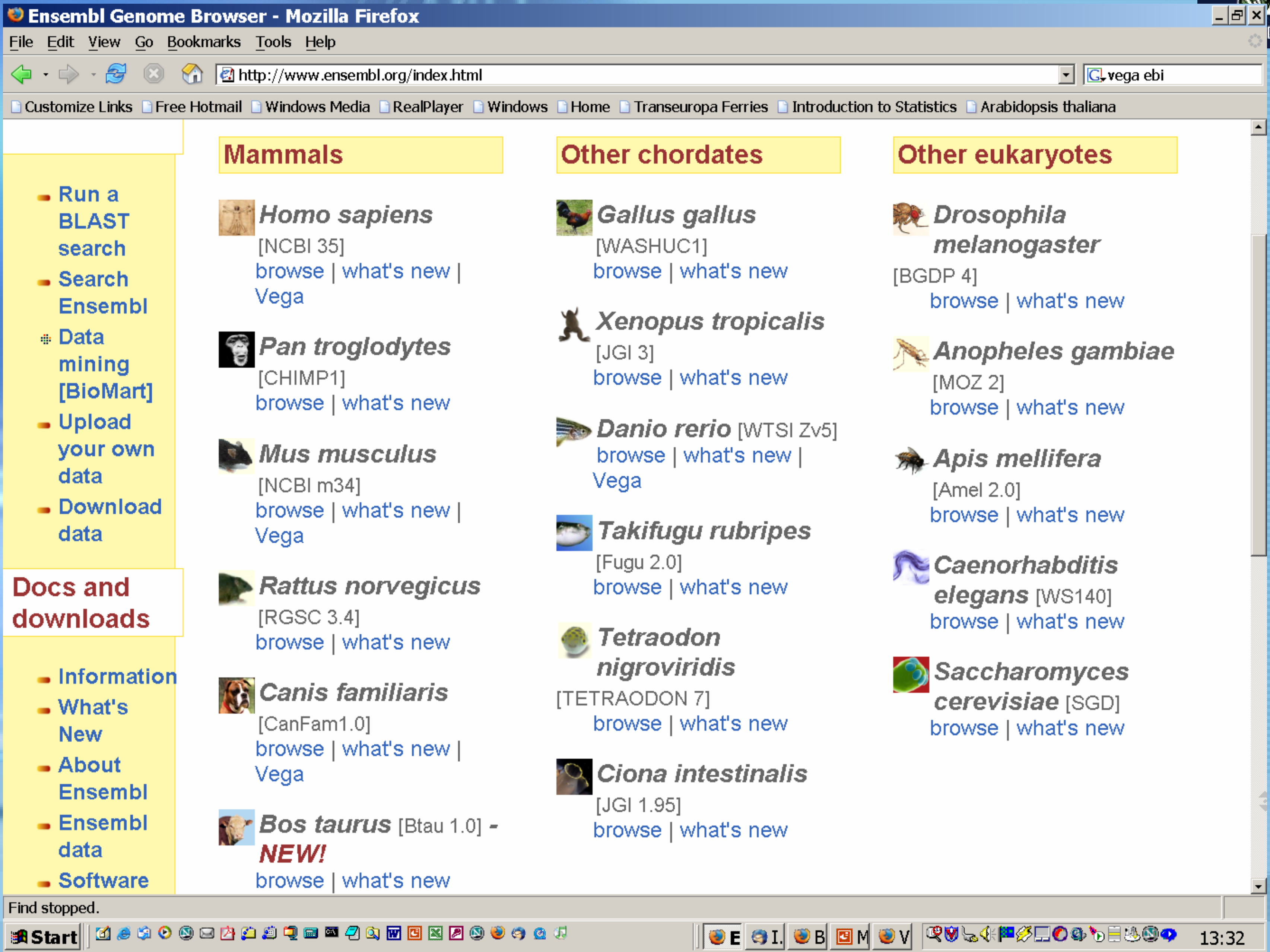
- Generic data management system aimed at complex interlinked datasets
- Collaboration between EBI and CSHL
- Originally developed for the Ensembl project but has now been generalized

http://www.biomart.org

Examples of BioMart databases

Ensembl

- Joint project between EMBL - EBI and the Sanger Institute
- Produces and maintains automatic annotation on selected eukaryotic genomes
- http://www.ensembl.org



- Run a BLAST search
- Search Ensembl
- Data mining [BioMart]
- Upload your own data
- Download data

Docs and downloads

- Information
- What's New
- About Ensembl
- Ensembl data
- Software

Mammals

Homo sapiens [NCBI 35]
browse | what's new | Vega

Pan troglodytes [CHIMP1]
browse | what's new

Mus musculus [NCBI m34]
browse | what's new | Vega

Rattus norvegicus [RGSC 3.4]
browse | what's new

Canis familiaris [CanFam1.0]
browse | what's new | Vega

Bos taurus [Btau 1.0] - ***NEW!***
browse | what's new

Other chordates

Gallus gallus [WASHUC1]
browse | what's new

Xenopus tropicalis [JGI 3]
browse | what's new

Danio rerio [WTSI Zv5]
browse | what's new | Vega

Takifugu rubripes [Fugu 2.0]
browse | what's new

Tetraodon nigroviridis [TETRAODON 7]
browse | what's new

Ciona intestinalis [JGI 1.95]
browse | what's new

Other eukaryotes

Drosophila melanogaster [BGDP 4]
browse | what's new

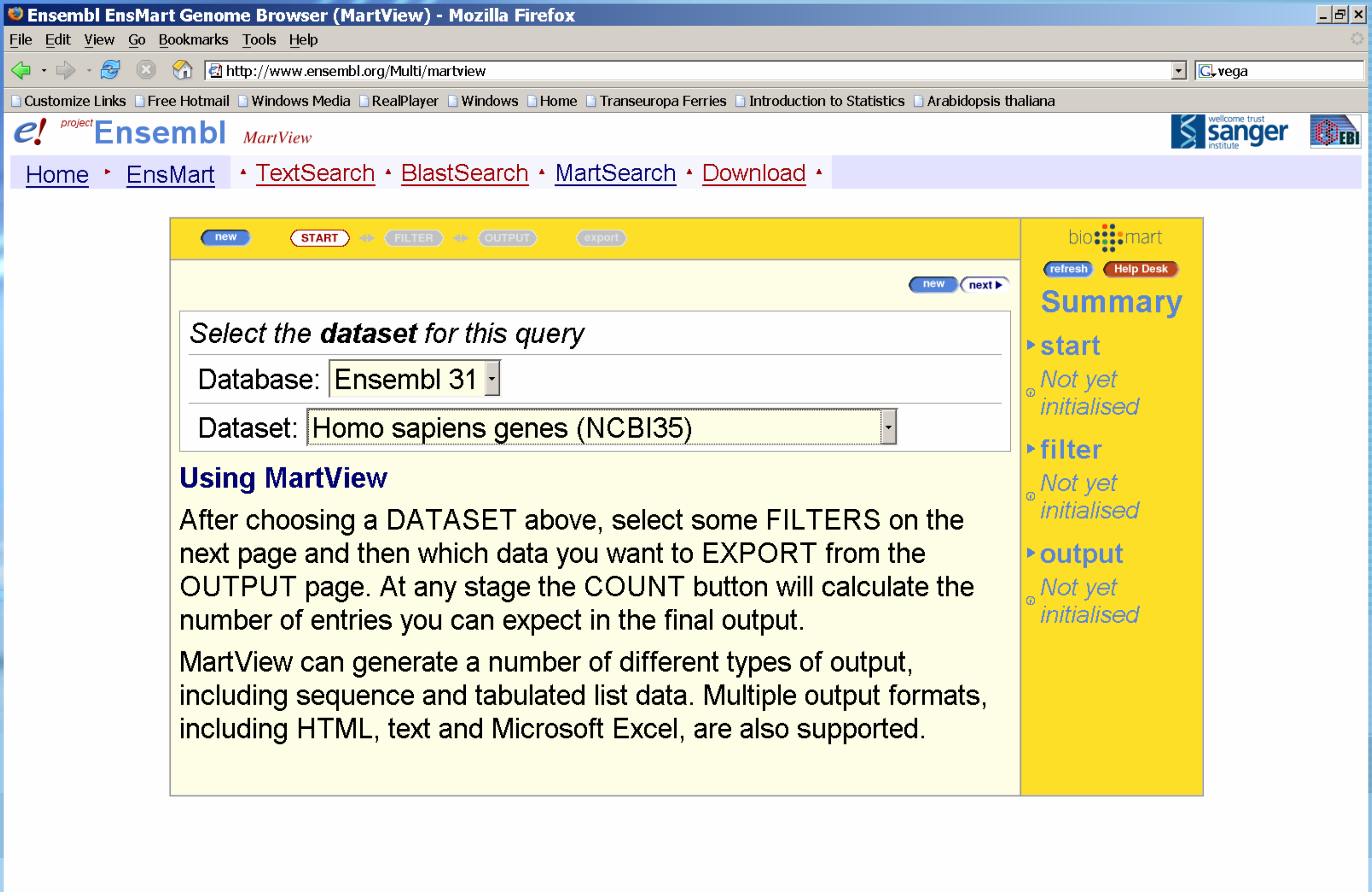
Anopheles gambiae [MOZ 2]
browse | what's new

Apis mellifera [Amel 2.0]
browse | what's new

Caenorhabditis elegans [WS140]
browse | what's new

Saccharomyces cerevisiae [SGD]
browse | what's new

Ensembl MartView


Ensembl EnsMart Genome Browser (MartView) - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.ensembl.org/Multi/martview
vega
Customize Links
Free Hotmail
Windows Media
RealPlayer
Windows
Home
Transeuropa Ferries
Introduction to Statistics
Arabidopsis thaliana
e! project Ensembl MartView
wellcome trust sanger institute
EBI
Home
EnsMart
TextSearch
BlastSearch
MartSearch
Download
new
START
FILTER
OUTPUT
export
new
next
biomart
refresh
Help Desk

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

Database: Ensembl 31

Dataset: Homo sapiens genes (NCBI35)

Using MartView

After choosing a DATASET above, select some FILTERS on the next page and then which data you want to EXPORT from the OUTPUT page. At any stage the COUNT button will calculate the number of entries you can expect in the final output.

MartView can generate a number of different types of output, including sequence and tabulated list data. Multiple output formats, including HTML, text and Microsoft Excel, are also supported.

Summary

start
Not yet initialised

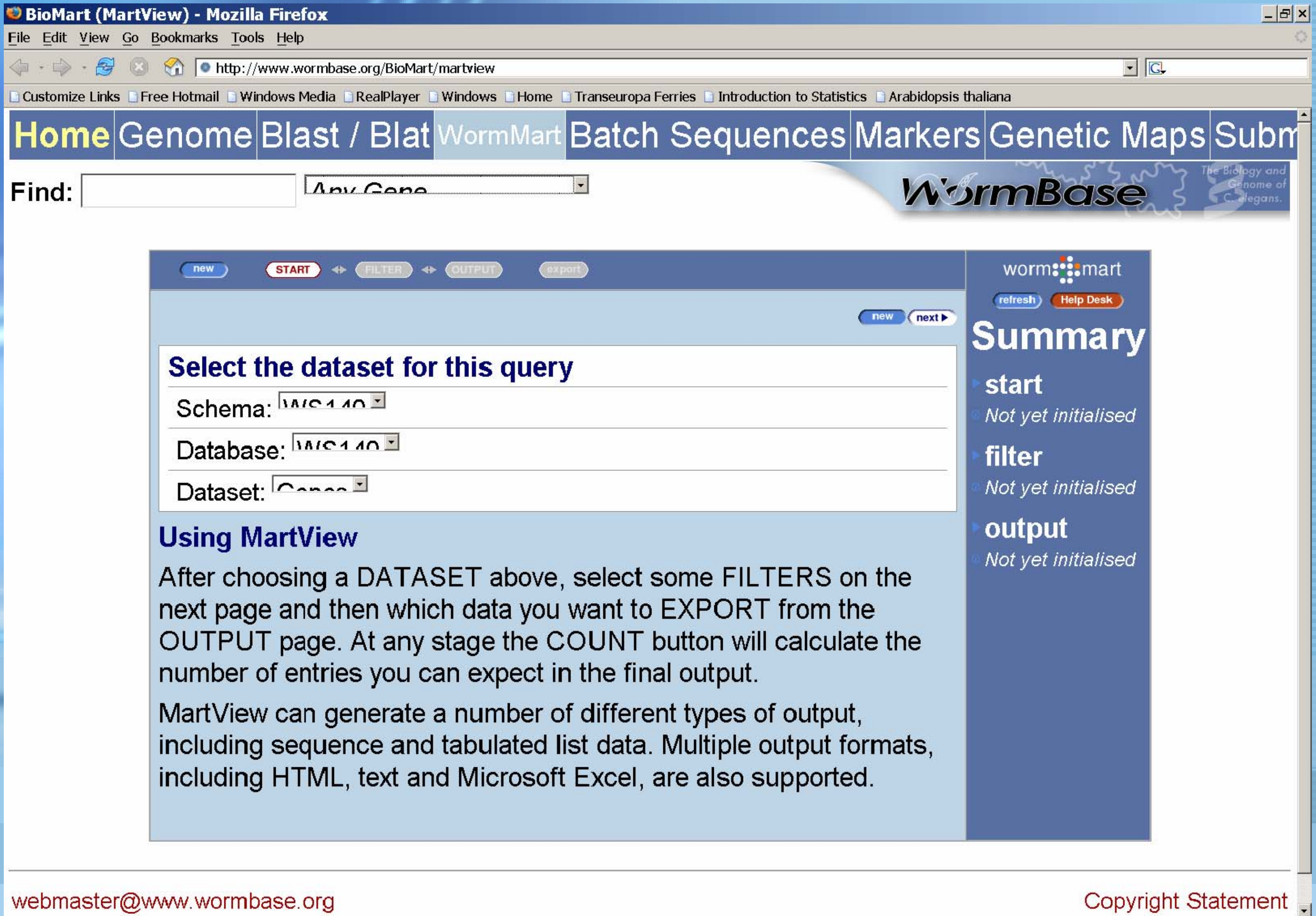
filter
Not yet initialised

output
Not yet initialised

Wormbase

- Repository of mapping, sequencing and phenotypic information on *C. elegans* (and some other nematodes)
- http://www.wormbase.org

WormMart



Select the dataset for this query

Schema: WS140

Database: WS140

Dataset: Genes

Using MartView

After choosing a DATASET above, select some FILTERS on the next page and then which data you want to EXPORT from the OUTPUT page. At any stage the COUNT button will calculate the number of entries you can expect in the final output.

MartView can generate a number of different types of output, including sequence and tabulated list data. Multiple output formats, including HTML, text and Microsoft Excel, are also supported.

Summary

start
Not yet initialised

filter
Not yet initialised

output
Not yet initialised

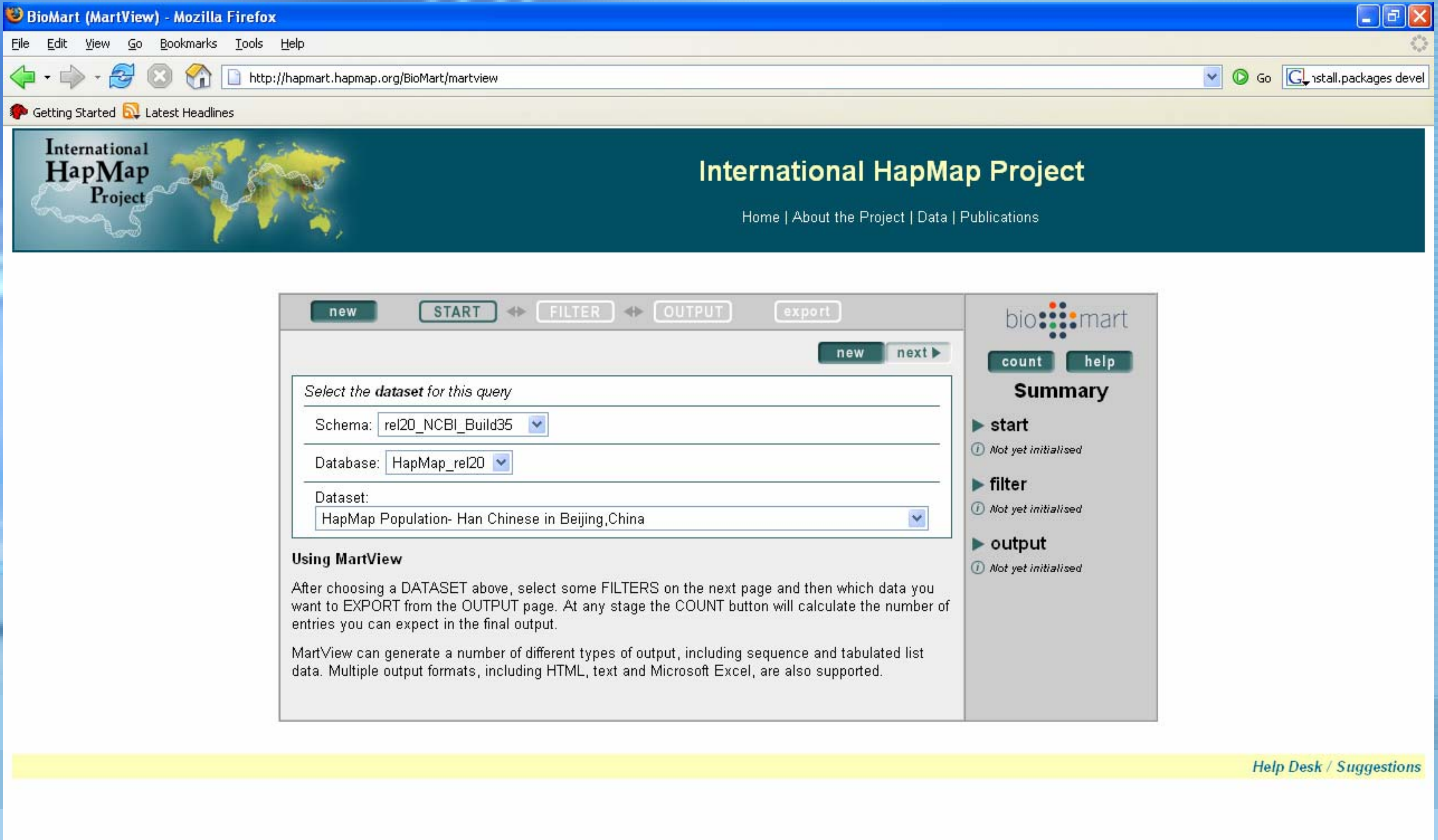
webmaster@www.wormbase.org

Copyright Statement

HapMap

- The HapMap Project is an international effort to identify and catalog genetic variation in human.

HapMap MartView



BioMart (MartView) - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://hapmart.hapmap.org/BioMart/martview
Go
install.packages devel
Getting Started
Latest Headlines
International HapMap Project

International HapMap Project

Home | About the Project | Data | Publications

new
START
FILTER
OUTPUT
export
new
next ▶
biomart
count
help

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

Schema: rel20_NCBI_Build35

Database: HapMap_rel20

Dataset: HapMap Population- Han Chinese in Beijing,China

Summary

▶ start
Not yet initialised

▶ filter
Not yet initialised

▶ output
Not yet initialised

Using MartView

After choosing a DATASET above, select some FILTERS on the next page and then which data you want to EXPORT from the OUTPUT page. At any stage the COUNT button will calculate the number of entries you can expect in the final output.

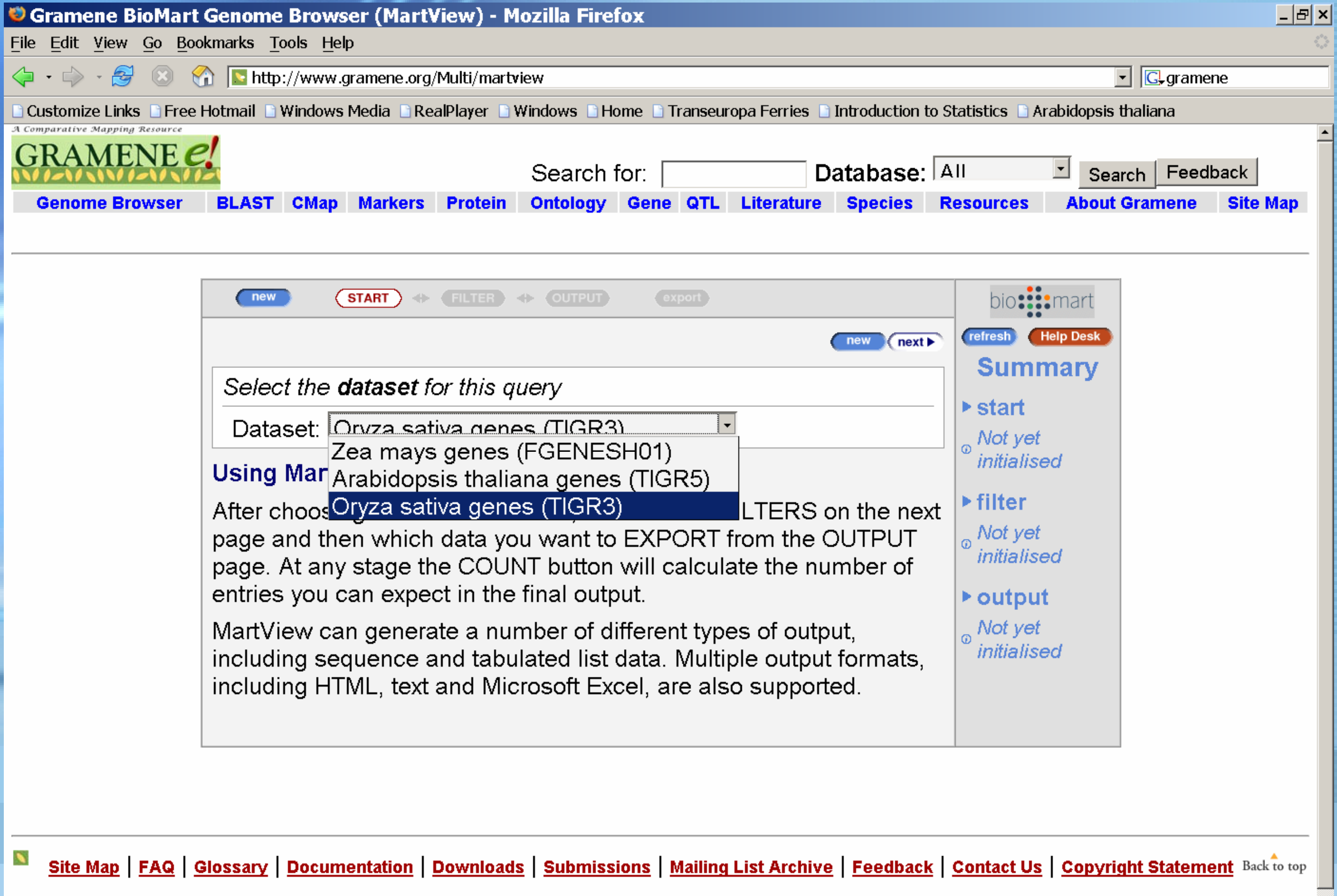
MartView can generate a number of different types of output, including sequence and tabulated list data. Multiple output formats, including HTML, text and Microsoft Excel, are also supported.

Help Desk / Suggestions

Gramene

- A comparative mapping resource for grains
- Includes:
  - Arabidopsis thaliana
- http://www.gramene.org

GrameneMart



Gramene BioMart Genome Browser (MartView) - Mozilla Firefox

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

Dataset: Oryza sativa genes (TIGR3)

- Zea mays genes (FGENESH01)
- Arabidopsis thaliana genes (TIGR5)
- Oryza sativa genes (TIGR3)

Using Mar

After choos LTERS on the next page and then which data you want to EXPORT from the OUTPUT page. At any stage the COUNT button will calculate the number of entries you can expect in the final output.

MartView can generate a number of different types of output, including sequence and tabulated list data. Multiple output formats, including HTML, text and Microsoft Excel, are also supported.

Summary

- start: *Not yet initialised*
- filter: *Not yet initialised*
- output: *Not yet initialised*

Site Map | FAQ | Glossary | Documentation | Downloads | Submissions | Mailing List Archive | Feedback | Contact Us | Copyright Statement
Back to top

Other publicly available databases with BioMart

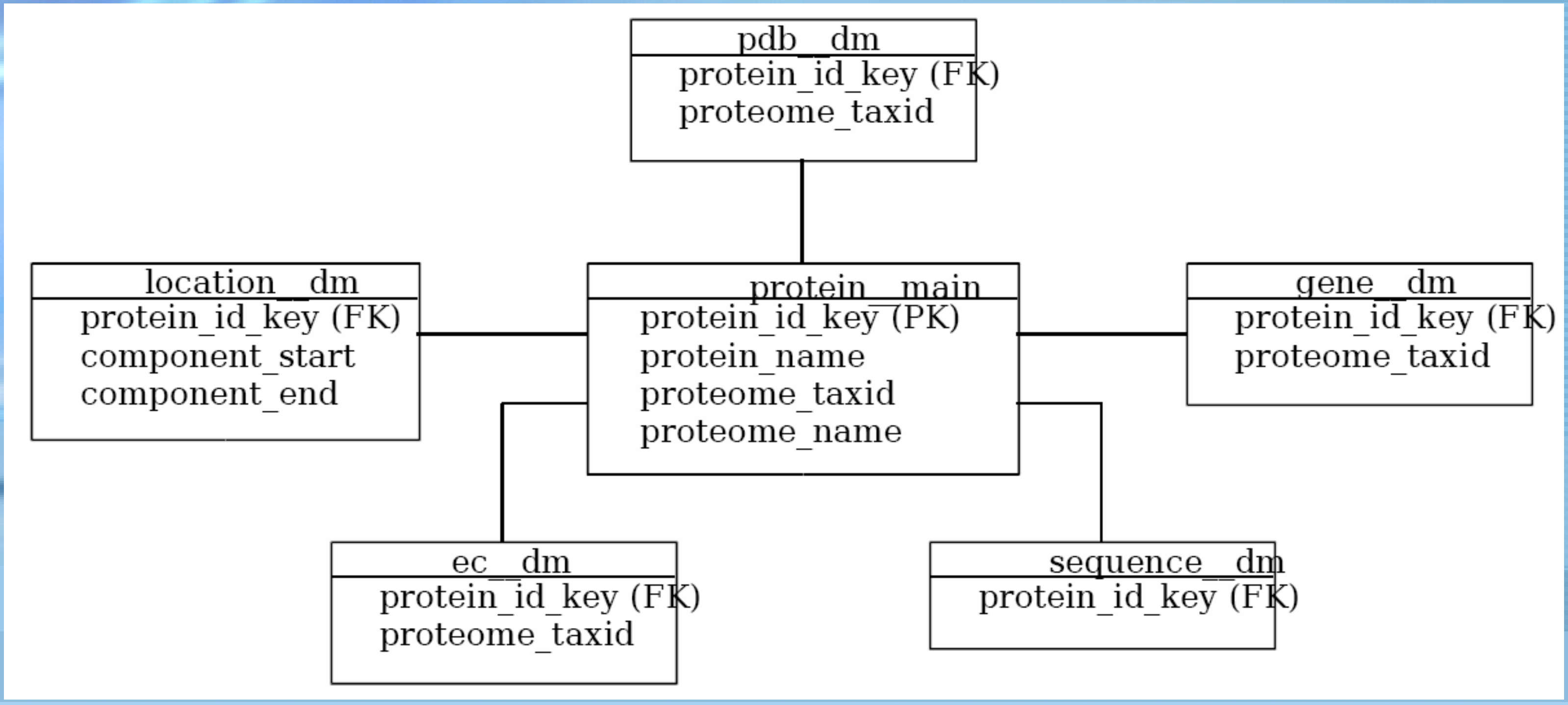
- euGenes
- VEGA
- Dictybase
- ZF-Models
- Uniprot
- MSD

BioMart databases

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

BioMart database schema’s

Simple star-like schema’s avoid complex joins and enable fast data retrieval



pdb__dm
protein_id_key (FK)
proteome_taxid

location__dm
protein_id_key (FK)
component_start
component_end

protein__main
protein_id_key (PK)
protein_name
proteome_taxid
proteome_name

gene__dm
protein_id_key (FK)
proteome_taxid

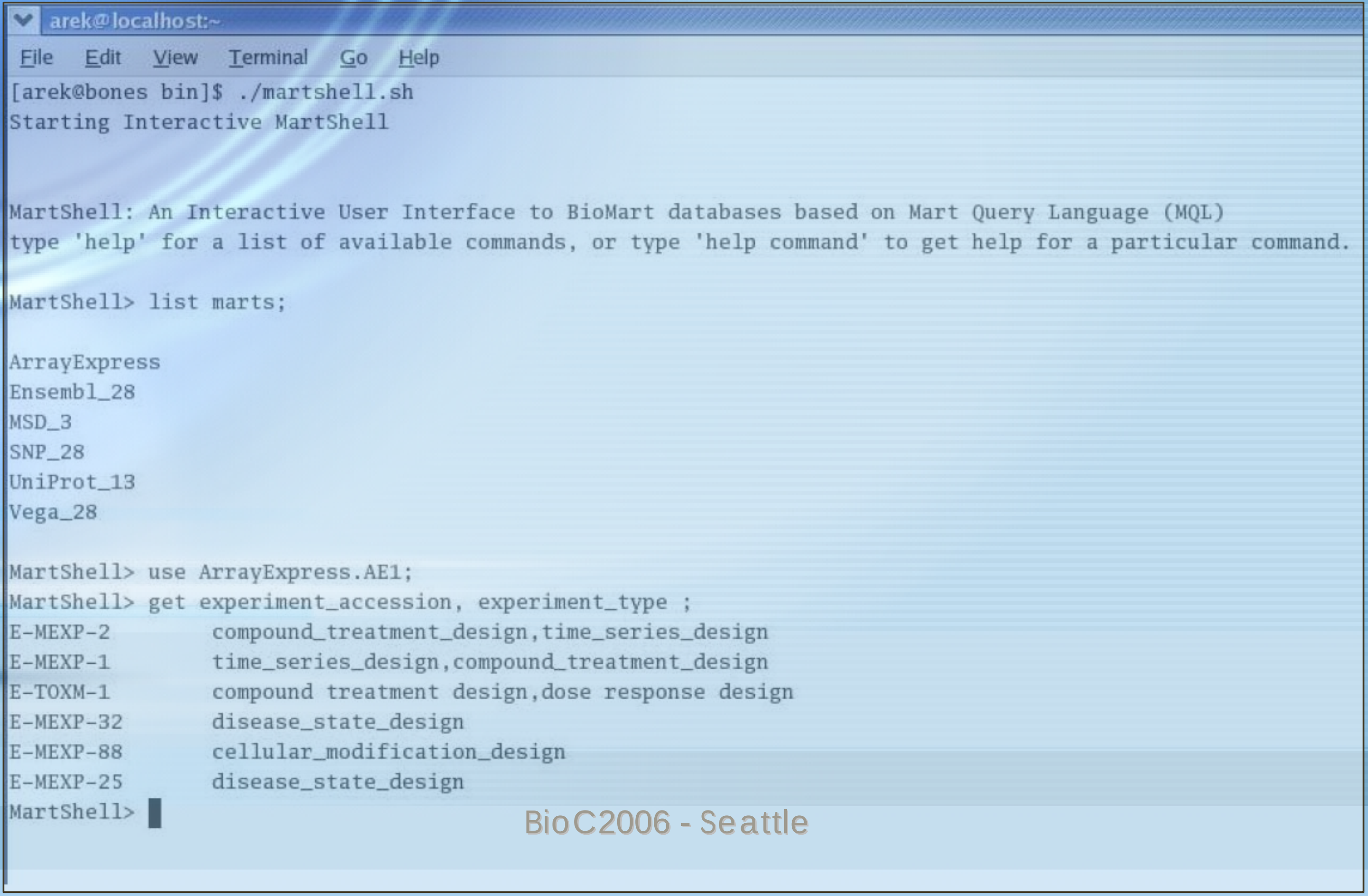
ec__dm
protein_id_key (FK)
proteome_taxid

sequence__dm
protein_id_key (FK)

BioMart user interfaces: MartShell

- Command-line BioMart user interface based on a structured query language: Mart Query Language (MQL)

BioMart user interfaces: MartShell



```
[arek@bones bin]$ ./martshell.sh
Starting Interactive MartShell


MartShell: An Interactive User Interface to BioMart databases based on Mart Query Language (MQL)
type 'help' for a list of available commands, or type 'help command' to get help for a particular command.

MartShell> list marts;

ArrayExpress
Ensembl_28
MSD_3
SNP_28
UniProt_13
Vega_28

MartShell> use ArrayExpress.AE1;
MartShell> get experiment_accession, experiment_type ;
E-MEXP-2        compound_treatment_design,time_series_design
E-MEXP-1        time_series_design,compound_treatment_design
E-TOXM-1        compound treatment design,dose response design
E-MEXP-32       disease_state_design
E-MEXP-88       cellular_modification_design
E-MEXP-25       disease_state_design
MartShell> █
```

BioMart user interfaces: MartView

- Web-based user interface for BioMart
- Provides functionality for remote users to query all databases hosted by the BioMart server

BioMart user interfaces: MartView

START Select BioMart and Dataset

FILTER Select a filter to restrict query e.g. Y chromosome

OUTPUT Select the output (attributes) e.g. entrezgene

BioMart user interfaces

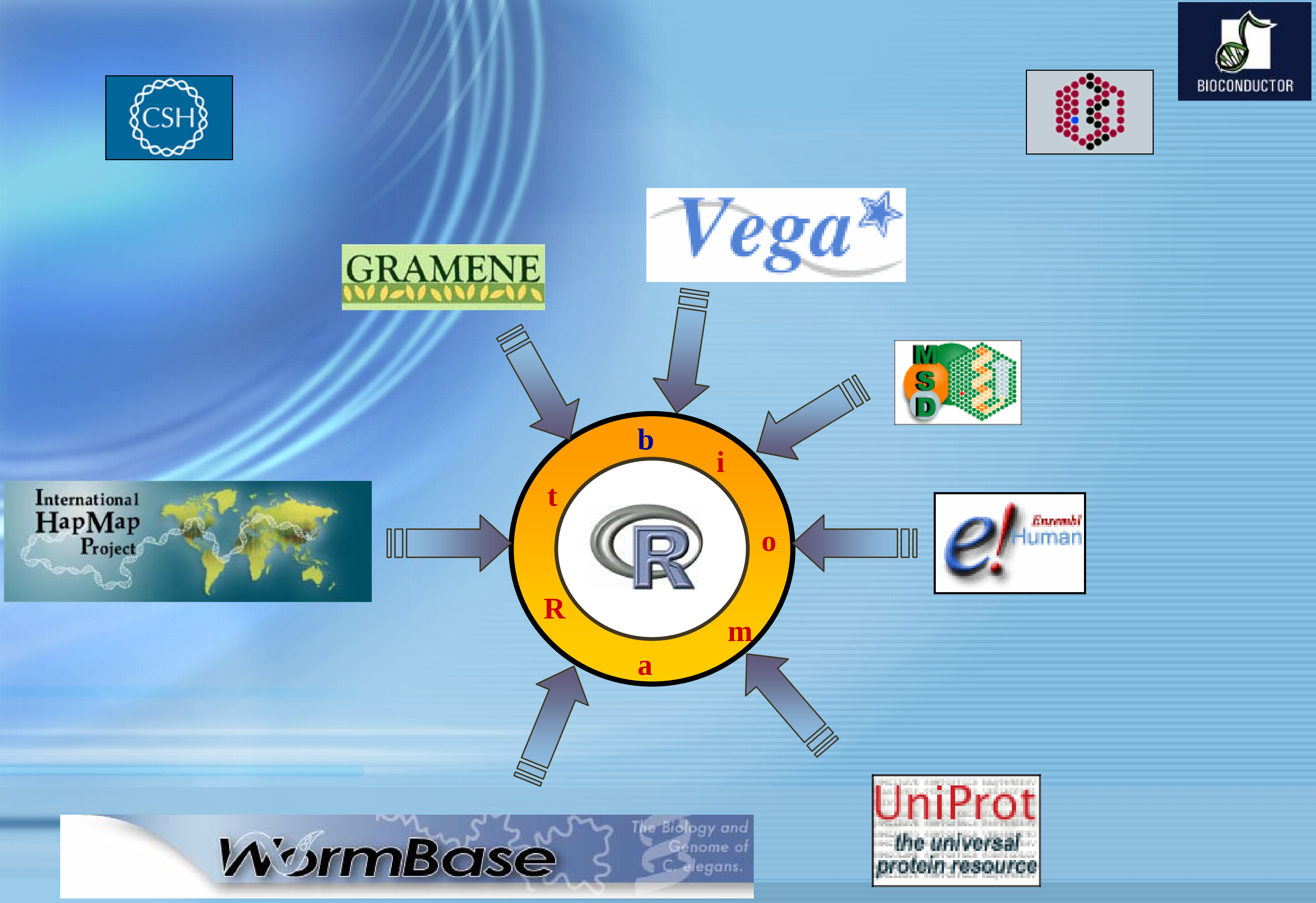
- MartExplorer - stand alone client
- Perl and Java libraries
- MartEditor

BioMart web service

- Web service:
  - A software system designed to support interoperable machine-to-machine interaction over a network.
  - Messages are typically conveyed using HTTP, and normally comprise XML in conjunction with other web-related standards

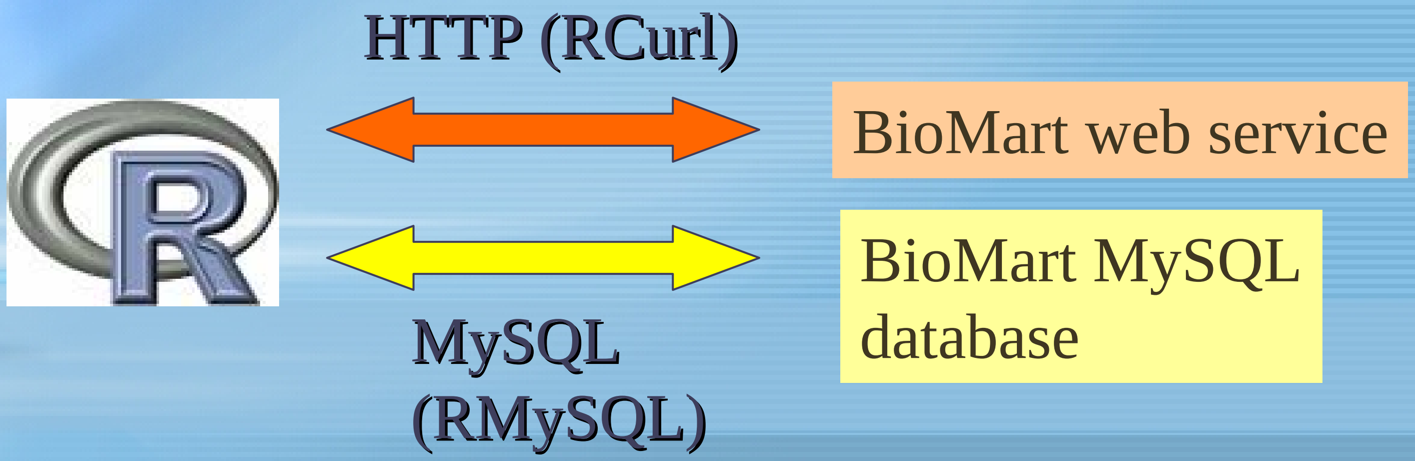
BIOCONDUCTOR

Integration of BioMart and R: biomaRt package


CSH
BIOCONDUCTOR
Vega
GRAMENE
MSD
International HapMap Project
e! Ensembl Human
b
i
o
m
a
R
t
R
WormBase
The Biology and Genome of C. elegans.
UniProt
the universal protein resource

biomaRt

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



HTTP (RCurl)

BioMart web service

BioMart MySQL database

MySQL (RMySQL)

biomaRt - Use

- Annotation of identifiers e.g. Affymetrix
- Retrieval of Gene Ontology, INTERPRO and other information
- Prioritizing groups of genes with particular properties
- Data mining

Hands-on

- Installation
- Selecting a BioMart database and dataset
- Simple biomaRt functions tailored to Ensembl
- Generic biomaRt functions

biomaRt installation

- biomaRt requires Rcurl
  http://www.omegahat.org/RCurl
- biomaRt requires XML package
- RMySQL package is optional
- Platforms on which biomaRt has been installed:
  - Linux (curl http://curl.haxx.se)
  - OSX (curl)
  - Windows

biomaRt installation

- Use biocLite

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

Selecting a BioMart

```
> library( biomaRt )
> listMarts()
```

Selecting a BioMart

```
$biomart
[1] "dicty"    "ensembl"  "snp"      "vega"     "uniprot"  "msd"
    "wormbase"

$version
[1] "DICTYBASE (NORTHWESTERN)"    "ENSEMBL 39 (SANGER)"
[3] "SNP 39 (SANGER)"             "VEGA 39 (SANGER)"
[5] "UNIPROT PROTOTYPE 4-5 (EBI)" "MSD PROTOTYPE 4 (EBI)"
[7] "WORMBASE CURRENT (CSHL)
```

Selecting a BioMart

```
> ensembl=useMart(“ensembl”)
```

Selecting a dataset

```
> listDatasets(ensembl)

         dataset                          version
1   rnorvegicus_gene_ensembl        RGSC3.4
2   scerevisiae_gene_ensembl        SGD1
3   celegans_gene_ensembl           CEL150
4   trubripes_gene_ensembl          FUGU4
5   cintestinalis_gene_ensembl      JGI2
6   ptroglodytes_gene_ensembl       CHIMP1A
7   agambiae_gene_ensembl           AgamP3
8   hsapiens_gene_ensembl           NCBI36
```

Selecting a dataset

```
> ensembl = useDataset(“hsapiens_gene_ensembl”, mart=ensembl)
```

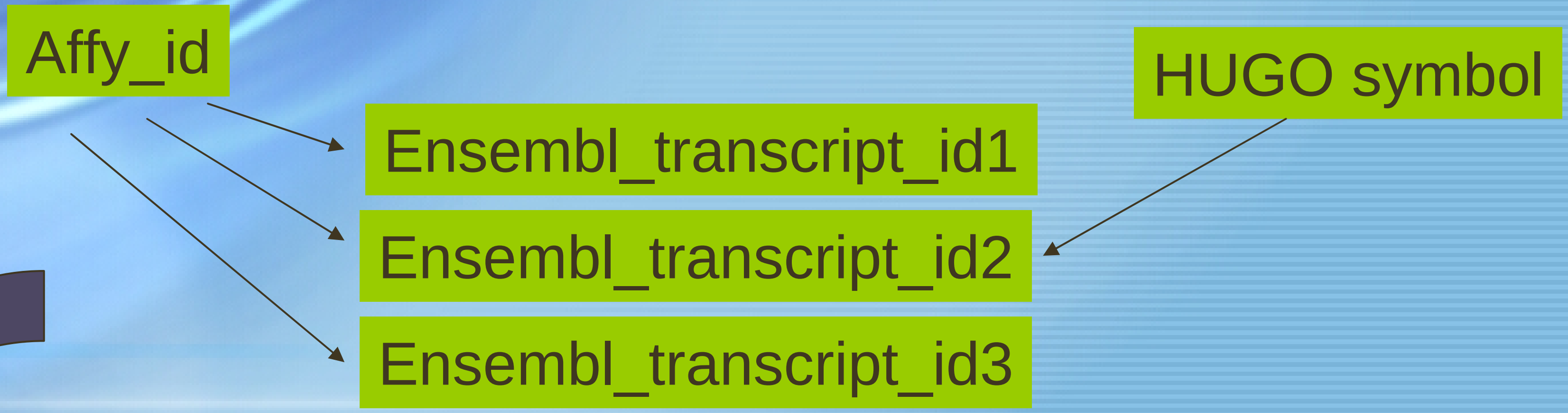
Or

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

Simple biomaRt functions tailored to Ensembl

Ensembl annotation

- Ensembl annotation is at the transcript level



1939_at	ENST000003789	
1939_at	ENST000003790	TP53
1939_at	ENST000003791	

getGene

Retrieves Gene annotations

- Gene symbol
- Description
- Chromosome name
- Band
- Start position
- End position
- Ensembl Gene ID
- Ensembl Transcript ID

getGene

- Annotation of many types of identifiers such as:
  - EntrezGene
  - Affymetrix
  - Refseq
  - Embl
  - ...

- Output of all biomaRt “get” functions is a data.frame

getGene

```
>  affyids = c("202763_at","209310_s_at",
                "207500_at")

> getGene(id=affyids,array="affy_hg_u133_plus_2",
                mart=ensembl)

        ID          symbol
 1   202763_at    CASP3
 2   207500_at    CASP5
 3 209310_s_at  CASP4
```

getGene

description

1 Caspase-3 precursor (EC 3.4.22.-) (CASP-3) (Apopain) (Cysteine protease CPP32) (Yama protein) (CPP-32) (SREBP cleavage activity 1) (SCA-1) [Contains: Caspase-3 p17 subunit; Caspase-3 p12 subunit]. [Source:Uniprot/SWISSPROT;Acc:P42574]

2 Caspase-5 precursor (EC 3.4.22.-) (CASP-5) (ICH-3 protease) (TY protease) (ICE(rel)-III) [Contains: Caspase-5 subunit p20; Caspase-5 subunit p10]. [Source:Uniprot/SWISSPROT;Acc:P51878]

3 Caspase-4 precursor (EC 3.4.22.-) (CASP-4) (ICH-2 protease) (TX protease) (ICE(rel)-II) [Contains: Caspase-4 subunit 1; Caspase-4 subunit 2]. [Source:Uniprot/SWISSPROT;Acc:P49662]

getGene

```
chromosome  band strand chromosome_start chromosome_end

1       4  q35.1      -1         185785845      185807623
2      11 q22.3      -1         104370180      104384909
3      11 q22.3      -1         104318810      104345373

ensembl_gene_id          ensembl_transcript_id

ENSG00000164305       ENST00000308394
ENSG00000137757       ENST00000260315
ENSG00000196954       ENST00000355546
```

getGene

Note:

- Ensembl does an independent mapping of affy probe sequences to genomes
- If there is no clear match then that probe is not assigned to a gene

getGO

- Retrieve Gene Ontology annotation for a list of identifiers
- Many identifiers can be used
- Returns GO id, GO description and evidence code

getGO

```
> getGO(id=affyids, array="affy_hg_u133_plus_2",mart=ensembl)

        ID      go_id             go_description                     evidence_code
1   202763_at GO:0005515   protein binding                                 IPI
2   202763_at GO:0008234   cysteine-type peptidase activity                IEA
3   202763_at GO:0030693   caspase activity                                TAS
4   202763_at GO:0006508   proteolysis                                     IDA
5   202763_at GO:0006915   apoptosis                                       IEA
6   202763_at GO:0006917   induction of apoptosis                          TAS
7   202763_at GO:0005737   cytoplasm                                        IDA
8   202763_at GO:0005737   cytoplasm                                       IEA
```

getINTERPRO

- INTERPRO is an integrated resource for protein families, domains and functional sites. It integrates secondary structure databases such as PROSITE, PRINTS, SMART, Pfam, ProDom, etc.
- Retrieve INTERPRO annotation for a list of identifiers
- Many identifiers can be used
- Returns INTERPRO id, description

getINTERPRO

```
> getINTERPRO(id=affyids[1],
  array="affy_hg_u133_plus_2",
  mart=ensembl)
```

getINTERPRO

```
        ID        interpro_id              description
1 202763_at   IPR001309   Caspase, p20 subunit
2 202763_at   IPR002398   Peptidase C14, caspase precursor p45
3 202763_at   IPR011600   Peptidase C14, caspase catalytic
4 202763_at   IPR002138 Peptidase C14, caspase non-catalytic
    subunit p10

          ensembl_gene_id          ensembl_transcript_id
1           ENSG00000164305       ENST00000308394
2           ENSG00000164305       ENST00000308394
3           ENSG00000164305       ENST00000308394
4           ENSG00000164305       ENST00000308394
```

Pre-selection of features

- Select all Affymetrix identifiers on the hgu133plus2 chip for genes located on chromosome 16 between base pair 1100000 and 1250000

```
> features = getFeature(
            array = "affy_hg_u133_plus_2“,
            chromosome = "16",
            start = "1100000",
            end="1250000", mart=ensembl)
```

Pre-selection of features

	ensembl_transcript_id	chromosome_name	start_position	end_position	affy_hg_u133_plus_
1	ENST00000358590	16	1143739	1211772	222960_at
2	ENST00000358590	16	1143739	1211772	205845_at
3	ENST00000356546	16	1143739	1211772	222960_at
4	ENST00000356546	16	1143739	1211772	205845_at
5	ENST00000234798	16	1211659	1215257	220339_s_at
6	ENST00000357113	16	1218338	1220215	207741_x_at

Pre-selection of features

```
> unique(features[,5])

"222960_at"   "205845_at"   "220339_s_at" "207741_x_at" "215382_x_at"
"210084_x_at" "205683_x_at" "207134_x_at" "217023_x_at" "216474_x_at"
"214568_at"
```

Pre-selection of features

- Select all entrezgene ids which have a "MAP kinase activity" GO term associated with it

```
> getFeature(type="entrezgene",
  GOID="GO:0004707", mart=ensembl)
```

Pre-selection of features

```
          GO        entrezgene
1  GO:0004707       5598
2  GO:0004707       5598
3  GO:0004707      51701
4  GO:0004707       5596
5  GO:0004707       5595
```

getSequence

- Retrieve sequences starting from a vector of identifiers or chromosomal coordinates
- 5’UTR
- 3’UTR
- cDNA
- protein

getSequence

```
> getSequence(chromosome=3,
        start=185514033,end=185535839,
        seqType="5utr", mart=ensembl)
```

```
CCGGCTGCGCCTGCGGAGAAGCGGTGGCCGCCGAGCGGGATCTGTGCGGGGAGCC
GGAAATGGTTGTGGACTACGTCTGTGCGGCTGCGTGGGGCTCGGCCGCGCGGACTG
AAGGAGACTGAAGGGGCGTTCCACATACGTTGTCCCGACACAGCAGTACCCTGTGC
AGCCAGGAGCCCCAGGCTTCTATCCAGGTGCAAGCCCTACAGAATTTGGGACCTAC
GCTGGCGCCTACTATCCAGCCCAAGGGGTGCAGCAGTTTCCCACTGGCGTGGCCCC
CACCCCAGTTTTG
```

getSequence

```
>getSequence(chromosome=3,
        start=185514033,end=185535839,
        seqType="cdna", mart=ensembl)
```

```
CCGGCTGCGCCTGCGGAGAAGCGGTGGCCGCCGAGCGGGATCTGTGCGGGGAGCC
GGAAATGGTTGTGGACTACGTCTGTGCGGCTGCGTGGGGCTCGGCCGCGCGGACTG
AAGGAGACTGAAGGGGCGTTCCACATACGTTGTCCCGACACAGCAGTACCCTGTGC
AGCCAGGAGCCCCAGGCTTCTATCCAGGTGCAAGCCCTACAGAATTTGGGACCTAC
GCTGGCGCCTACTATCCAGCCCAAGGGGTGCAGCAGTTTCCCACTGGCGTGGCCCC
CACCCCAGTTTTGATGAACCAGCCACCCCAGATTGCTCCCAAGAGGGAGCGTAAGA
CGATCCGAATTCGAGATCCAAACCAAGGAGGAAAGGATATCACAGAGGAGATCATG
TCTGGGGCCCGCACTGCCTCCACACCCACCCCTCCCC..........
```

getSequence

```
>getSequence(chromosome=3,
    start=185514033,end=185535839,
    seqType="peptide", mart=ensembl)
```

```
MNQPPQIAPKRERKTIRIRDPNQGGKDITEEIMSGARTASTPT
PPQTGGGLEPQANGETPQVAVIVRPDDRSQGAIIADRPGLPG
PEHSPSESQPSSPSPTPSPSPVLEPGSEPNLAVLSIPGDTMT
TI.......
```

getSNP

- SNP: Single Nucleotide Polymorphisms, are common DNA sequence variations among individuals
- dbSNP is mirrored by Ensembl in its snp BioMart.
- getSNP retrieves tsc-ids and refsnp identifiers together with allele, chromosome start and strand information.

getSNP

```
> snpmart = useMart("snp", dataset = "hsapiens_snp")

> snp=getSNP(chromosome = 8, start = 148350, end = 148612,  mart = snpmart)
```

```
       tscid  refsnp_     id      allele chrom_start chrom_strand
1  TSC1723456  rs3969741    C/A      148394            1
2  TSC1421398  rs4046274    C/A      148394            1
3  TSC1421399  rs4046275    A/G      148411            1
4              rs13291      C/T      148462            1
5  TSC1421400  rs4046276    C/T      148462            1
6              rs4483971    C/T      148462            1
7              rs17355217   C/T      148462            1
8              rs12019378   T/G      148471            1
9  TSC1421401  rs4046277    G/A      148499            1
10             rs11136408   G/A      148525            1
11 TSC1421402  rs4046278    G/A      148533            1
12             rs17419210   C/T      148533           -1
13             rs28735600   G/A      148533            1
14 TSC1737607  rs3965587    C/T      148535            1
15             rs4378731    G/A      148601            1
```

getHomolog

- 18 different species in Ensembl are interlinked
- biomaRt takes advantage of this to provide homology mappings between different species
- Combine two Ensembl datasets

getHomolog

```
> human = useMart("ensembl","hsapiens_gene_ensembl")
> mouse = useMart("ensembl","mmusculus_gene_ensembl")
```

getHomolog

```
> homolog = getHomolog( id = "1939_at",
                        to.array = "affy_mouse430_2",
                        from.array = "affy_hg_u95av2",
                        from.mart = human,
                        to.mart = mouse )
> homolog
                  V1                 V2           V3
1 ENSMUSG00000059552 ENSMUST00000005371 1427739_a_at
2 ENSMUSG00000059552 ENSMUST00000005371 1426538_a_at
```

getHomolog

```
> homolog = getHomolog( id = "NM_007294",
                        to.array = "affy_mouse430_2",
                        from.type = "refseq",
                        from.mart = human,
                        to.mart = mouse )

> homolog
                  V1                 V2           V3
1 ENSMUSG00000017146 ENSMUST00000017290  1424629_at
2 ENSMUSG00000017146 ENSMUST00000017290   1451417_at
3 ENSMUSG00000017146 ENSMUST00000017290 1424630_a_at
```

Generic biomaRt queries

Generic biomaRt queries

- Previous functions were all tailored to Ensembl BioMart
- Generic functions can be used to any available BioMart database and are modeled after MQL
- Generic functions enable one to query everything that is made available by the database

Filters

- Filters define restrictions on the query
- Conceptually filters are inputs

- Example filters:
  - entrezgene
  - chromosome_name

listFilters

- Returns vector of all filters available in the selected BioMart

```
> listFilters(ensembl)
  [1] "affy_hc_g110"                  "affy_hg_focus"
  [3] "affy_hg_u133_plus_2"               "affy_hg_u133a"
  [5] "affy_hg_u133a_2"                 "affy_hg_u133b"
……
 [15] "agilent_probe"                   "biotype"
 [17] "ccds"                      "chromosome_name"
 [19] "embl"                      "end"
…….
```

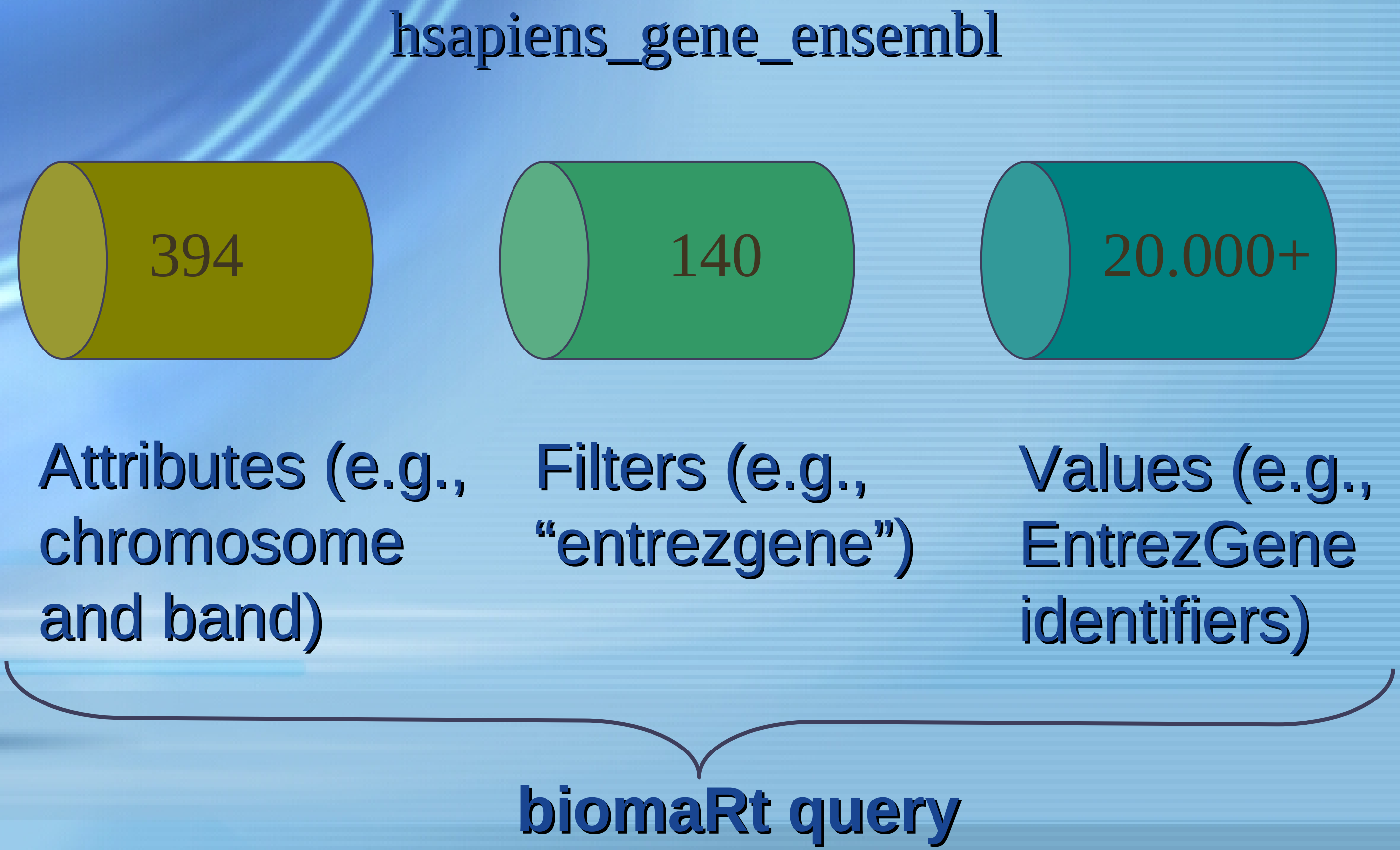
Attributes

- Attributes define the values which the user is interested in.
- Conceptually equal to output of the query
- Example attributes:
  - chromosome_name
  - band

listAttributes

```
> listAttributes(ensembl)
  [1] "adf_embl"
         .......
 [14] "affy_hg_u95av2"
 [15] "affy_hg_u95b"
         .......
 [21] "agilent_cgh"
 [22] "agilent_probe"
 [23] "allele"
 [24] "allele_frequency"
 [25] "band"
        .....
```

Generic biomaRt queries



hsapiens_gene_ensembl

394

140

20.000+

Attributes (e.g., chromosome and band)

Filters (e.g., “entrezgene”)

Values (e.g., EntrezGene identifiers)

biomaRt query

getBM

- Generic biomaRt query function
- Contains no hard-coded information
- Is used by the simple biomaRt functions (which do contain hard-coded names of attributes and filters)

getBM

```
> getBM(
attributes=c("affy_hg_u95av2","hgnc_symbol",
"chromosome_name","band"),
filters="affy_hg_u95av2",
values=c("1939_at","1454_at"), mart=mart)
```

```
  affy_hg_u95av2 hgnc_symbol chromosome_name
    band
1        1454_at      SMAD3              15 q22.33
2        1939_at       TP53           17  p13.1
```

getBM: homology mapping

- Within one Ensembl dataset there are attributes providing mappings to the other Ensembl species

- Example:

  starting from the hsapiens dataset and a list of entrezgene ids we can query chromosomal positions of the corresponding genes in human, zebrafish, mouse and mosquito.

getBM: homology mapping

```
> getBM(attributes=
 c("hgnc_symbol","chromosome_name","start_position",
    "mouse_chromosome","mouse_chrom_start",
    "zebrafish_chromosome","zebrafish_chrom_start",
    "mosquito_chromosome","mosquito_chrom_start"),

   filter="entrezgene",
   values = c("673","7157","837"),
   mart=ensembl)
```

```
  hgnc_symbol chromosome_name start_position
     BRAF              7          140080754
     TP53             17            7512464
    CASP4             11          104318810
mouse_chromosome mouse_chrom_start
               6          39543731
              11          69396600
               9           5308874
zebrafish_chromosome zebrafish_chrom_start
                   4               9473158
                   5              16155000
                  16              47717138

  mosquito_chromosome mosquito_chrom_start
                   2L              1974599
                   2R             20538788
                                        NA
```

Using Wormbase

```
> listMarts()

$biomart
[1] "dicty"     "ensembl"  "snp"       "vega"      "uniprot"
    "msd"       "wormbase"

$version
[1] "DICTYBASE (NORTHWESTERN)"     "ENSEMBL 39
    (SANGER)"
[3] "SNP 39 (SANGER)"             "VEGA 39 (SANGER)"
[5] "UNIPROT PROTOTYPE 4-5 (EBI)" "MSD
    PROTOTYPE 4 (EBI)"
[7] "WORMBASE CURRENT (CSHL)
```

Using Wormbase

```
>wormbase=useMart("wormbase",dataset="gene")
> listFilters(wormbase)
> listAttributes(wormbase)
>getBM(attributes=c("name","rnai",
                    "rnai_phenotype","phenotype_desc"),
                    filters="gene_name",
                  values=c("unc-26","his-33"),
                     mart=wormbase)
```

Using Wormbase

```
 name  rnai              rnai_phenotype
   phenotype_desc

1  his-33 WBRNAi00000104   Emb | Nmo
   embryonic lethal | Nuclear morphology alteration in
   early embryo
2  his-33 WBRNAi00012233   WT
   wild type morphology
3  his-33 WBRNAi00024356   Ste
   sterile
4  his-33 WBRNAi00025036   Emb
   embryonic lethal
```

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

Reporting bugs

- Check with MartView if you get the same output
  - Yes: contact database e.g.

    helpdesk@ensembl.org

  - No: contact me

Acknowledgements

- EBI
  - Wolfgang Huber
  - Arek Kasprzyk
  - Ewan Birney
  - Alvis Brazma
- ESAT-SCD KULeuven
  - Yves Moreau

- NIH/NHGRI

  -Sean Davis

- Bioconductor users