Database mining with biomaRt

Steffen Durinck

NIH/NCI -> Lawrence Berkeley National Laboratory

BioC 2007

Overview

- The BioMart software suite
- biomaRt package
- biomaRt installation
- biomaRt example queries to show the variety of different data types/questions that can be retrieved/answered for many organisms

BioMart 0.6

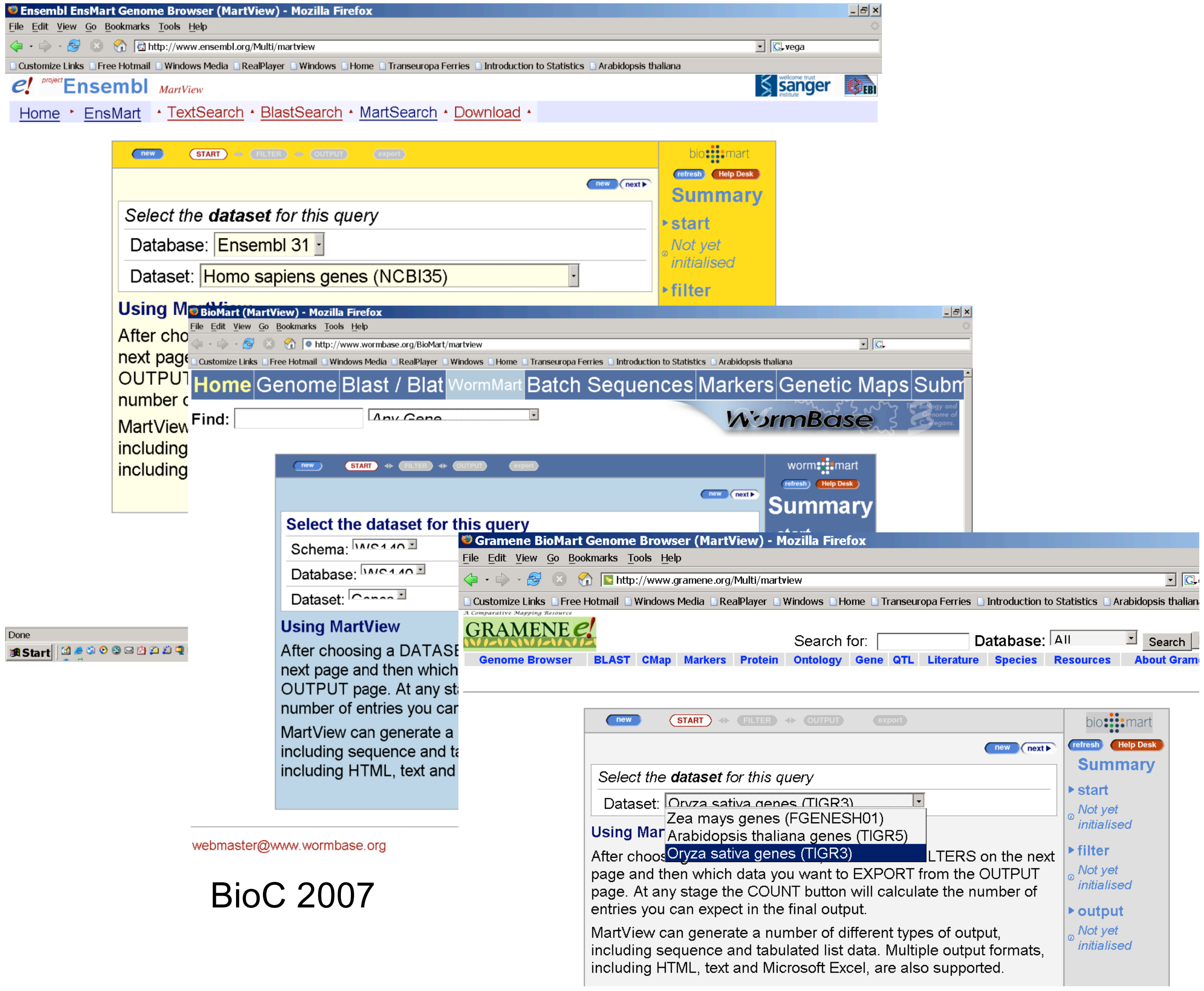
- BioMart is a query-oriented data management system developed jointly by the European Bioinformatics Institute (EBI) and Cold Spring Harbor Laboratory (CSHL).
- Originally developed for the Ensembl project but has now been generalized

BioMart 0.6

- BioMart data can be accessed using either web, graphical, or text based applications, or programmatically using web services or software libraries written in Perl and Java.
- http://www.biomart.org

Example BioMart databases

- Ensembl
- Wormbase
- Uniprot
- Gramene
- HapMap


Ensembl EnsMart Genome Browser (MartView) - Mozilla Firefox

http://www.ensembl.org/Multi/martview

Ensembl MartView

Home • EnsMart • TextSearch • BlastSearch • MartSearch • Download •

Select the **dataset** for this query

Database: Ensembl 31

Dataset: Homo sapiens genes (NCBI35)

Summary

start

Not yet initialised

filter

BioMart (MartView) - Mozilla Firefox

http://www.wormbase.org/BioMart/martview

Home Genome Blast / Blat WormMart Batch Sequences Markers Genetic Maps Subm

Find: Any Gene

Select the dataset for this query

Schema: WS140

Database: WS140

Dataset: Genes

Using MartView

After choosing a DATASE next page and then which OUTPUT page. At any st number of entries you car

MartView can generate a including sequence and ta including HTML, text and

webmaster@www.wormbase.org

Gramene BioMart Genome Browser (MartView) - Mozilla Firefox

http://www.gramene.org/Multi/martview

Search for: Database: All Search

Genome Browser BLAST CMap Markers Protein Ontology Gene QTL Literature Species Resources About Gram

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

Example BioMart databases

- VEGA
- MSD
- Dictybase

- To come:
  - Reactome
  - …

BioMart databases

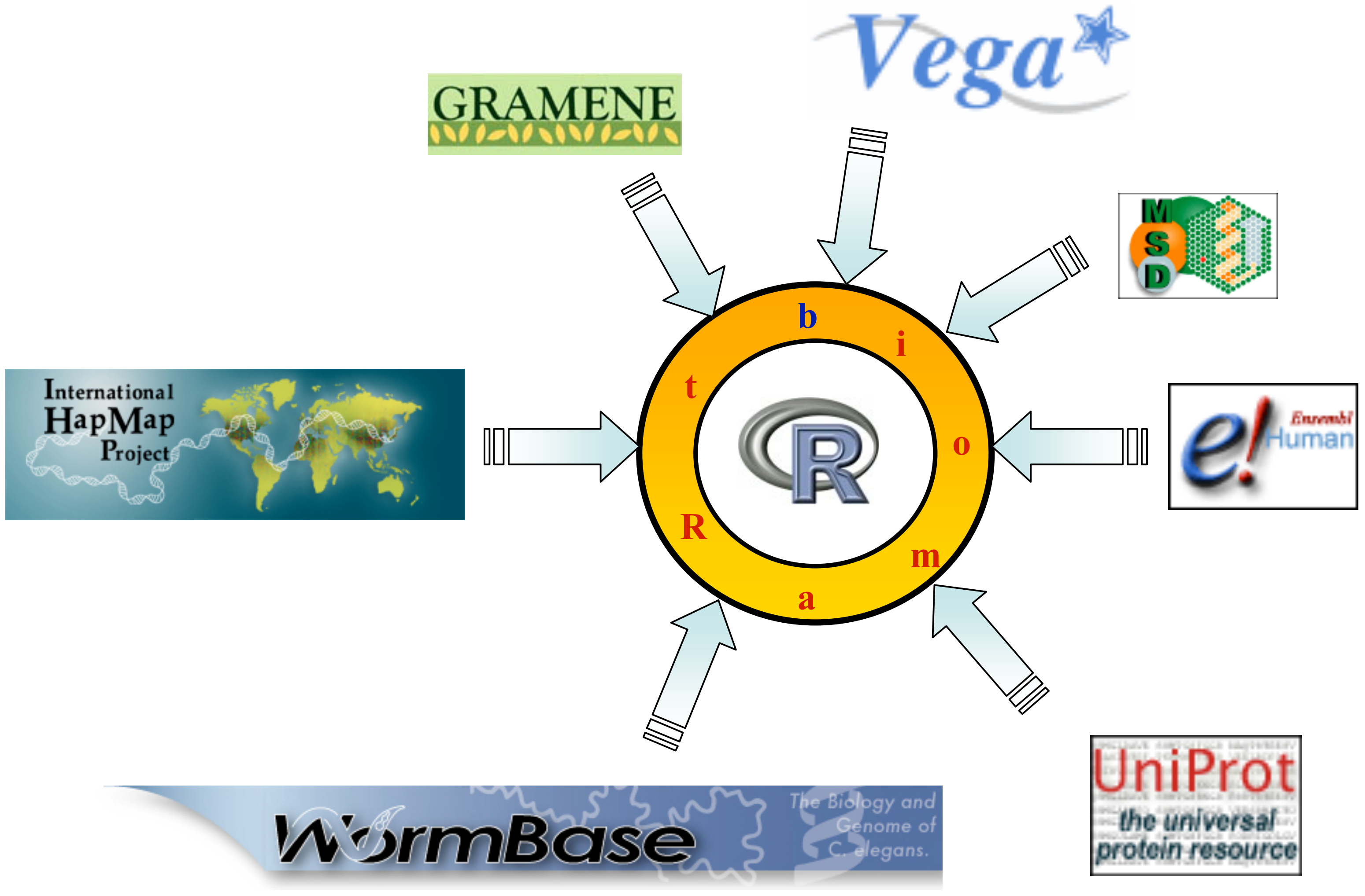
- De-normalized
- Tables with ‘redundant’ information
- Query optimized
- Fast and flexible

- Well suited for batch querying

biomaRt

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

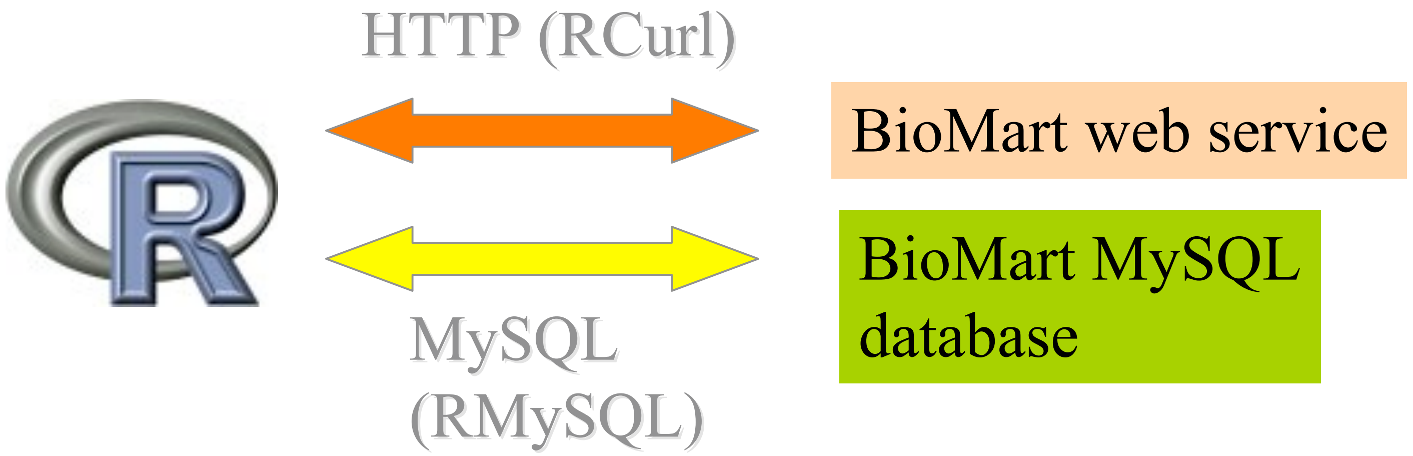
biomaRt - aim



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

biomaRt - db access

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



Installing biomaRt

- Platforms on which biomaRt has been installed:
  - Linux (curl http://curl.haxx.se)
  - OSX (curl)
  - 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

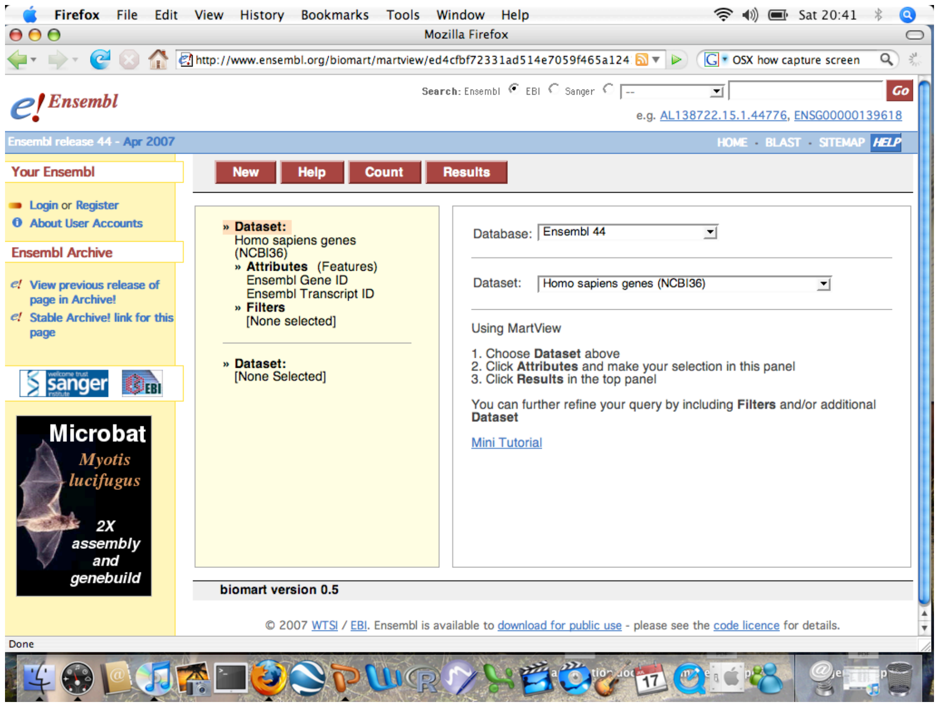
1	ensembl	ENSEMBL 45 GENES (SANGER)
2	compara_mart_homology_45	ENSEMBL 45 HOMOLOGY
3	compara_mart_pairwise_ga_45	ENSEMBL 45 PAIRWISE ALIGNMENTS
4	compara_mart_multiple_ga_45	ENSEMBL 45 MULTIPLE ALIGNMENTS
5	snp	ENSEMBL 45 VARIATION (SANGER)
6	vega	VEGA 21 (SANGER)
7	uniprot	UNIPROT PROTOTYPE (EBI)
8	msd	MSD PROTOTYPE (EBI)
9	ENSEMBL_MART_ENSEMBL	GRAMENE (CSHL)
10	wormbase176	WORMBASE (CSHL)
11	dicty	DICTYBASE (NORTHWESTERN)
12	rgd_mart	RGD GENES (MCW)
13	SSLP_mart	RGD MICROSATELLITE MARKERS (MCW)
14	pepseekerGOLD_mart	PEPSEEKER
15	pride	PRIDE (EBI)
16	Pancreatic_Expression	PANCREATIC EXPRESSION DATABASE

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 - BioMart

> ensembl=useMart("ensembl")


Firefox File Edit View History Bookmarks Tools Window Help
Sat 20:41
Mozilla Firefox
http://www.ensembl.org/biomart/martview/ed4cfbf72331ad514e7059f465a124
OSX how capture screen
Ensembl
Search: Ensembl EBI Sanger
Go
e.g. AL138722.15.1.44776, ENSG00000139618
Ensembl release 44 - Apr 2007
HOME · BLAST · SITEMAP HELP
Your Ensembl
Login or Register
About User Accounts
Ensembl Archive
View previous release of page in Archive!
Stable Archive! link for this page
sanger
EBI
Microbat
Myotis lucifugus
2X assembly and genebuild
New
Help
Count
Results
Done

» **Dataset:**
Homo sapiens genes
(NCBI36)
» **Attributes** (Features)
Ensembl Gene ID
Ensembl Transcript ID
» **Filters**
[None selected]

» **Dataset:**
[None Selected]

Database: Ensembl 44

Dataset: Homo sapiens genes (NCBI36)

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

© 2007 WTSI / EBI. Ensembl is available to download for public use - please see the code licence for details.

Ensembl - Datasets

```
> listDatasets(ensembl)
```

Returns:

- name: *hsapiens_gene_ensembl*
- description: *Homo sapiens genes*
- version: *NCBI36*

Ensembl currently contains 38 datasets~species

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

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

biomaRt query: Filters

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

- Example filters:
  - entrezgene
  - chromosome_name

biomaRt query

Attributes (e.g., chromosome and band)

Filters (e.g., “entrezgene”)

Values (e.g., EntrezGene identifiers)

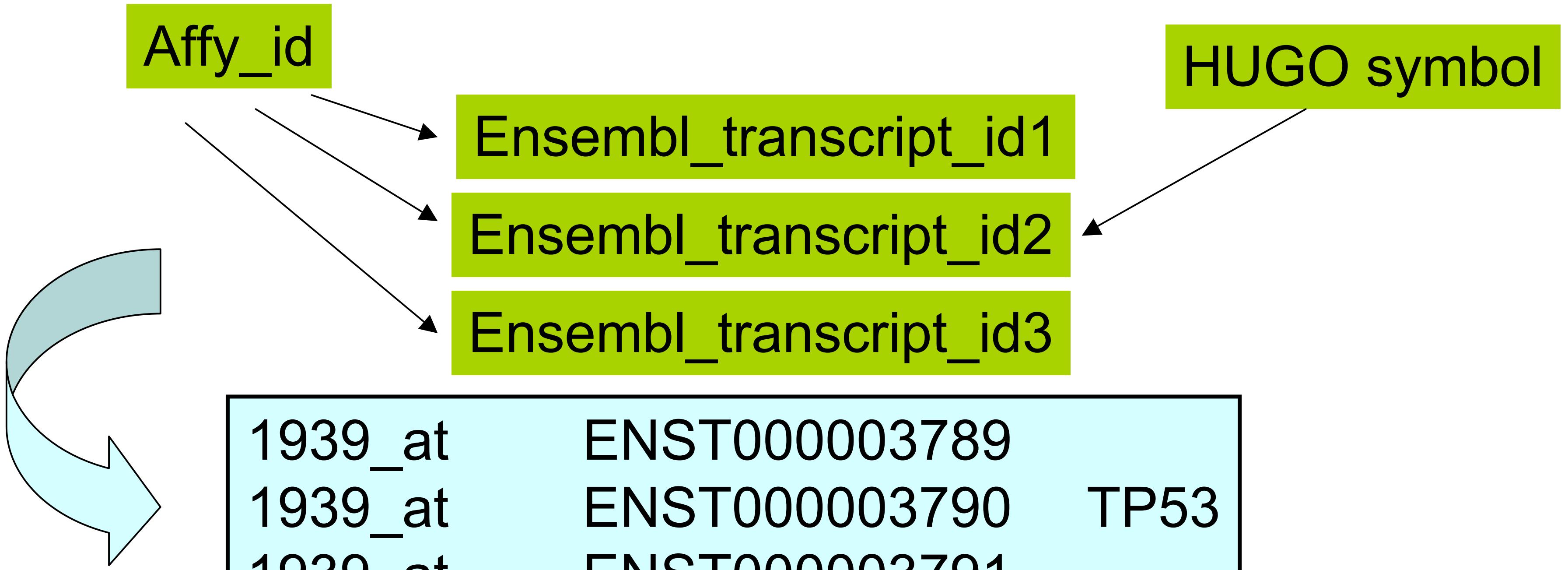
biomaRt query

Three main biomaRt functions

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

Ensembl annotation

- Ensembl annotates everything on the transcript level



1939_at	ENST000003789	
1939_at	ENST000003790	TP53
1939_at	ENST000003791	

Affymetrix & Ensembl

- 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

TASK 1 - Ensembl

- Annotate the following Affymetrix probe identifiers from the human u133plus2 platform with hugo gene nomenclature symbol (hgnc_symbol) and chromosomal location information:

211550_at, 202431_s_at, 206044_s_at

TASK 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

TASK 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)
```

TASK 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
```

TASK 1* - Ensembl

Retrieve GO annotation for the same set of affy ids

```
> annotation =
  getBM(attributes=c("affy_hg_u133_plus_2","ense
  mbl_transcript_id","ensembl_gene_id",”go”,
  ”go_description”), filters="affy_hg_u133_plus_2",
  values=affyids,mart = ensembl)
```

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

TASK 2 - Ensembl

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

TASK 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

TASK 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)
> diab
  ensembl_gene_id hgnc_symbol
1 ENSG00000139515        PDX1
2 ENSG00000139515        PDX1
3 ENSG00000108753        TCF2
4 ENSG00000108753        TCF2
5 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)

Boolean filters/ *filterType*

The function *filterType* allows you to figure out which type each filter is (this function is currently only available in the devel version of biomaRt)

```
> filterType("affy_hg_u133_plus_2", mart=ensembl)
[1] "text"
```

```
>filterType("with_affy_hg_u133_plus_2", mart=ensembl)
[1] "boolean"
```

TASK 3 - Ensembl

Retrieve all miRNAs known on chromosome 13 and their chromosomal locations

TASK 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
```

TASK 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
```

attributeSummary/filterSummary

- attributeSummary gives brief overview of available attribute categories and groups

```
> attributeSummary(ensembl)
   category                    group
1   Features                 EXTERNAL:
2   Features                   GENE:
3   Features               EXPRESSION:
4   Features                 PROTEIN:
5   Features            GENOMIC REGION:
6   Homologs            AEDES ORTHOLOGS:
7   Homologs        ANOPHELES ORTHOLOGS:
8   Homologs        ARMADILLO ORTHOLOGS:
9   Homologs         BUSHBABY ORTHOLOGS:
```

attributeSummary/filterSummary

- listFilters function can now show specific subset only e.g. SNP’s

```
> listAttributes(ensembl, category=”SNPs")
                 name                    description
1               allele                   Allele
2         chromosome_location            Chromosome Location (bp)
3             external_id                Reference ID
4             fpcctg_name                fpcctg name
5           gene_location Location in Gene (coding etc)
6               hgbase                   HGBASE ID
7              mapweight                 Mapweight
8    non_synonymous_snp_count            Non-synonymous SNP count
```

Additional help to figure out which filter and attribute names to use

- Go to www.biomart.org and select BioMart you use
- Select attributes and filters
- Press to XML button to get their names

TASK 4 - Ensembl

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

TASK 4 - Ensembl

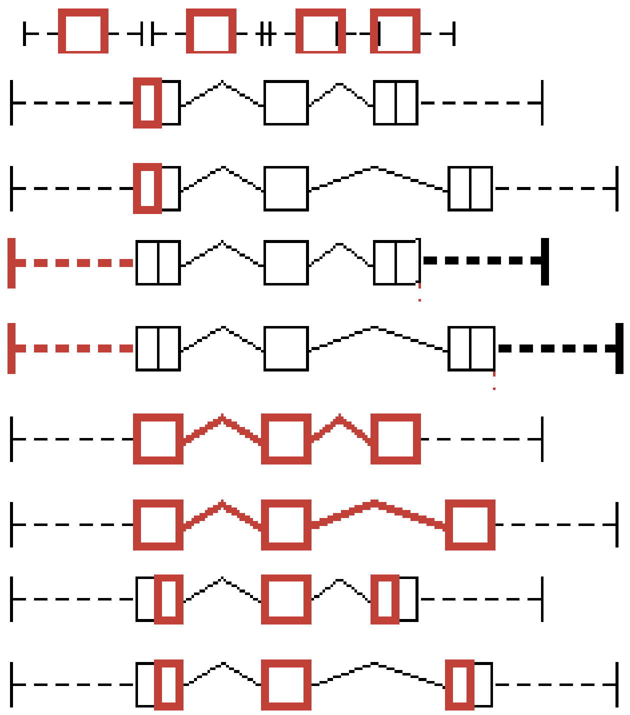
```
> entrez =
   getBM("entrezgene",filters=c("chromosome_name”
   ,"with_coding_snp"),
   values=list(22,TRUE),mart=ensembl)
> entrez[1:5,]
>  entrez[1:5,]
[1] 649486  81061 440153 150160 150165
```

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

TASK 5 - Ensembl

Retrieve all exons of CDH1

TASK 5 - 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
```

TASK 6 - 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'

TASK 6 - 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
```

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

TASK 7 - Ensembl

Retrieve human gene symbol and affy identifiers of their homologs in chicken for the following two identifiers from the human affy_hg_u95av2 platform: 976_s_at, 1888_s_at

TASK 7 - 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
4 1888_s_at          KIT         Gga.606.1.S1_at
```

TASK 8 - Ensembl

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

TASK 8 - Ensembl

```
> mouse=useMart("ensembl", dataset="mmusculus_gene_ensembl")
  Checking attributes and filters ... ok
> human=useMart("ensembl", dataset="hsapiens_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
```

SNP BioMart

- dbSNP mapped to Ensembl

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

TASK 9 - SNP

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

TASK 9 - SNP

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

TASK 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 (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.

TASK 10 - Uniprot

Which is the longest annotated protein in human?

TASK 10 - Uniprot

```
> lengths = getBM(c("protein_name","length"),
   filters=c("proteome_name","length_greater"),
   values=list("Homo sapiens",400),mart=unip)
> longest = which(lengths[,2] == max(lengths[,2]))
> lengths[longest,]
      protein_name length
4832         Titin    34350
```

TASK 11 - Uniprot

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

TASK 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
```

TASK 12 - Uniprot

Determine the INTERPRO protein domains of PDGFRA

TASK 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

TASK 13 - Gramene

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

TASK 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)
```

TASK 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.

TASK 14 - Wormbase

Determine the RNAi ids and the observed phenotypes for the gene with wormbase gene id: WBGene00006763

TASK 14 - Wormbase

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

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

TASK 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
```

biomaRt wrapper functions for Ensembl

- A set of frequently used queries to Ensembl are provided as wrapper functions in biomaRt.

- *getGene*: annotation of list of identifiers with symbol, chromosome name, band, start and end position, strand
- *getGO*: Retrieves GO id and description starting from list of identifiers
- *getSNP*: retrieval of refSNP identifiers given a chromosomal region

biomaRt wrapper functions

- *getHomolog*: Maps identifers of one species to identifiers in other species
- *getFeature*: retrieves set of identifiers given chromosomal location or GO id

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 - sdurinck@gmail.com

Acknowledgements

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

- Bioconductor users