Workshop

DATABASE MINING WITH BIOMART

Steffen Durinck
Lawrence Berkeley National Laboratory

BioC2010 - Seattle

Overview

- Brief intro:
  - The BioMart software suite
  - biomaRt package
  - biomaRt installation
- Tour of BioMart databases available through biomaRt
  - Example queries to show the variety of different data types/questions that can be retrieved/ answered for many organisms

BioMart 0.7

- 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 0.7

- 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

bio mart

HOME | MARTVIEW | MARTSERVICE | DOCS | CONTACT | NEWS | CREDITS

BioMart Project

BioMart is a query-oriented data management system developed jointly by the Ontario Institute for Cancer Research (OICR) and the European Bioinformatics Institute (EBI).

The system can be used with any type of data and is particularly suited for providing 'data mining' like searches of complex descriptive data. BioMart comes with an 'out of the box' website that can be installed, configured and customised according to user requirements. Further access is provided by graphical and text based applications or programmatically using web services or API written in Perl and Java. BioMart has built-in support for query optimisation and data federation and in addition can be configured to work as a DAS 1.5 Annotation server. The process of converting a data source into BioMart format is fully automated by the tools included in the package. Currently supported RDBMS platforms are MySQL, Oracle and Postgres.

BioMart is completely Open Source, licensed under the LGPL, and freely available to anyone without restrictions.

Powered by BioMart software:

- BioMart Central Portal
- ICGC Data Portal
- Ensembl
- Ensembl Bacteria
- Ensembl Metazoa
- Ensembl Protists
- Ensembl Plants
- Ensembl Fungi
- Phytozome
- Gramene
- Europhenome
- UniProt
- InterPro
- HGNC
- Rice-Map
- IKMC
- Wormbase
- DroSpeGe
- ArrayExpress DW
- Eurexpress
- HapMap
- Dictybase
- COSMIC
- IntOGen
- Rat Genome Database
- GermOnLine
- PRIDE
- PepSeeker
- VectorBase
- HTGT
- Cildb
- Pancreatic Expression Database
- Reactome
- EU Rat Mart
- Paramecium DB
- International Potato Center (CIP)
- Mouse Genome Informatics (MGI)
- CyanoBase

Third party software with BioMart Plugin:

Bioclipse | biomaRt-BioConductor | Cytoscape | Galaxy | Gitools | Ruby API | Taverna | WebLab

BioMart databases

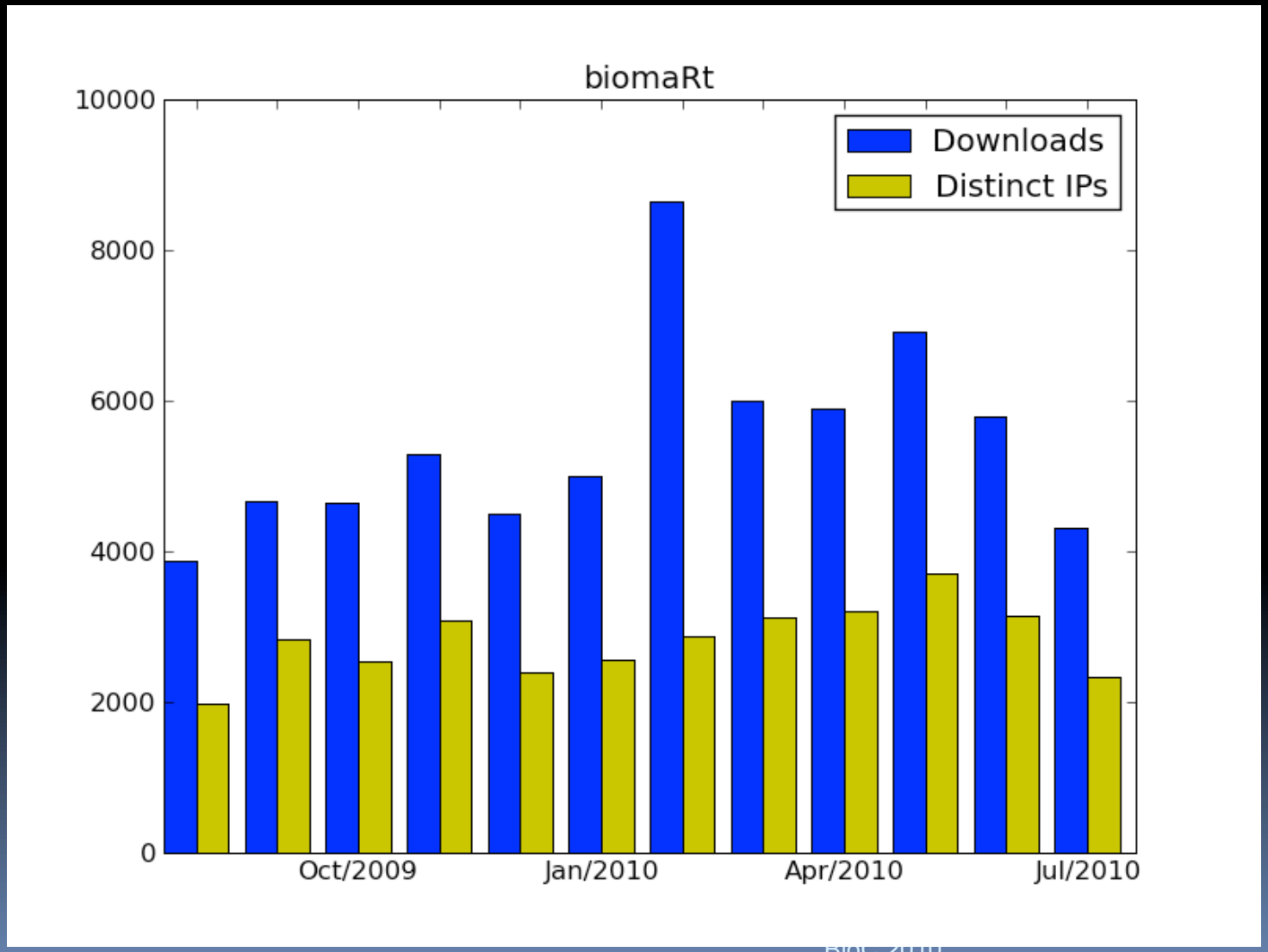
- De-normalized
- Tables with 'redundant' information
- Query optimized
- Fast and flexible

- Well suited for batch querying

biomaRt

- R interface to BioMart databases
- Performs online queries
- Current release version 2.4.0
- Imports Rcurl and XML packages

Package Download Stats



Installing biomaRt

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

Wiki with code example for this tutorial

http://biomart2010.wikispaces.com/

Installing biomaRt

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

> biocLite('biomaRt')

Running biocinstall version 2.4.11 with R version 2.9.1

Your version of R requires version 2.4 of Bioconductor.

also installing the dependencies 'bitops', 'XML', 'RCurl', 'biomaRt'

List available BioMart databases

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

List available BioMarts (currently 42 BioMarts)

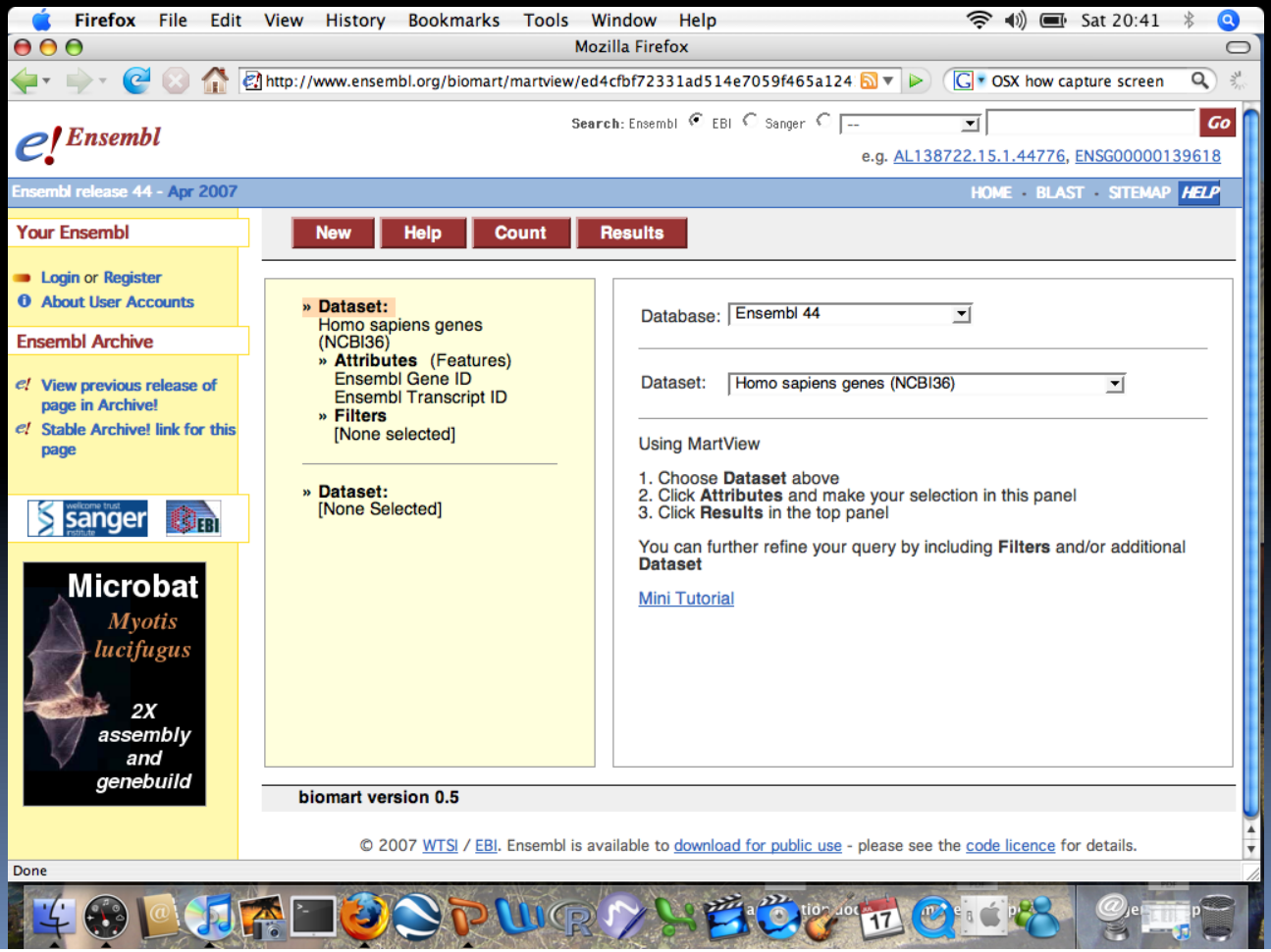
	biomart	version
1	ensembl	ENSEMBL GENES 58 (SANGER UK)
2	snp	ENSEMBL VARIATION 58 (SANGER UK)
3	functional_genomics	ENSEMBL FUNCTIONAL GENOMICS 58 (SANGER UK)
4	vega	VEGA 38 (SANGER UK)
5	bacterial_mart_5	ENSEMBL BACTERIA 5 (EBI UK)
6	fungal_mart_5	ENSEMBL FUNGAL 5 (EBI UK)
7	.....	

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")*



Ensembl - Datasets

> *listDatasets(ensembl)*

Returns:

- name: *hsapiens_gene_ensembl*
- description: *Homo sapiens genes*
- version: *(GRCh37)*

Ensembl currently contains 50 datasets~species

Ensembl - Datasets

A dataset can be selected using the useMart function

```
> ensembl = useMart("ensembl",
    dataset="hsapiens_gene_ensembl)
```

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

biomaRt query: 3 parts

Attributes (e.g., chromosome and band)

Filters (e.g., entrezgene)

Values (e.g., list of entrezgene ids)

biomaRt query

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

Three main biomaRt functions

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

Microarrays & Ensembl

- Ensembl does an independent mapping of array probe sequences to genomes (Affymetrix, Illumina, Agilent,…)
- 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","ensembl_gene_id","hgnc_sy
  mbol","chromosome_name","start_position","end_posi
  tion","band","strand"), filters="affy_hg_u133_plus_2",
  values=affyids,
 mart = ensembl)
```

TASK 1 - Ensembl

```
>annotation
 affy_hg_u133_plus_2 ensembl_gene_id hgnc_symbol chromosome_name
1       202431_s_at  ENSG00000136997       MYC           8
2       206044_s_at  ENSG00000157764      BRAF           7
3        211550_at   ENSG00000146648      EGFR           7

  start_position end_position   band strand
     128748316   128753671             q24.21     1
     140433817   140624564             q34         -1
      55086714    55324313  p11.2       1
```

TASK 1* - Ensembl

Retrieve GO annotation for the following Illumina human_wg6_v2 identifiers:

ILMN_1728071, ILMN_1662668

```
> illuminaIDs = c("ILMN_1728071","ILMN_1662668")
> goAnnot = getBM(c("illumina_humanwg_6_v2",
  "go_biological_process_id","go_biological_process_link
  age_type"), filters="illumina_humanwg_6_v2",
  values=illuminaIDs, mart = ensembl)
```

TASK 1* - Ensembl

	illumina_humanwg_6_v2	go_biological_process_id	go_biological_process_linkage_type
1	ILMN_1662668	GO:0000281	IMP
2	ILMN_1662668	GO:0006461	IDA
3	ILMN_1662668	GO:0006974	IDA
4	ILMN_1662668	GO:0007026	IDA
5	ILMN_1662668	GO:0007050	IDA

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(c("ensembl_gene_id","hgnc_symbol"),
           filters=c("mim_morbid_accession","go"),
           values=list(c("125853","222100"),"GO:0003700"),
           mart=ensembl)
```

TASK 2 - Ensembl

	ensembl_gene_id	hgnc_symbol
1	ENSG00000139515	PDX1
2	ENSG00000108753	HNF1B
3	ENSG00000148737	TCF7L2
4	ENSG00000106331	PAX4
5	ENSG00000162992	NEUROD1
6	ENSG00000135100	HNF1A

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] "id_list

>filterType("with_affy_hg_u133_plus_2", mart=ensembl)

[1] "boolean_list"

TASK 3 - Ensembl

Retrieve all miRNAs known on chromosome 13 and their chromosomal locations

TASK 3 - Ensembl

```
>miRNA = getBM(c
    ("mirbase_id","ensembl_gene_id","start_position",
"chromosome_name"), filters=c
    ("chromosome_name","with_mirbase"), values=list(13,TRUE),
    mart=ensembl)
> miRNA[1:5,]
```

TASK 3 - Ensembl

```
> miRNA[1:5,]
    mirbase_id    ensembl_gene_id      start_position    chromosome_name
1   hsa-mir-622 ENSG00000207858      90883436          13
2   hsa-mir-19a ENSG00000207610      92003145          13
3  hsa-mir-92a-1 ENSG00000207968      92003568          13
4   hsa-mir-18a ENSG00000199180      92002997          13
5 hsa-mir-320d-1 ENSG00000211491      41301964          13
```

attributePages

- attributePages gives brief overview of available attribute pages (useful for displaying subset of attributes)

```
> attributePages(ensembl)
[1] "feature_page" "structure"    "snp"          "homologs"     "sequences"

>listAttributes(ensembl, page = "feature_page" )
```

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

FilterOptions function: enumerates all possible values for a filter (if available)

TASK 4 - Ensembl

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

TASK 4 - Ensembl

> filterOptions("snptype_filters",ensembl)

[1] "[STOP_GAINED,STOP_LOST,COMPLEX_INDEL,FRAMESHIFT_CODING,

***NON_SYNONYMOUS_CODING**,STOP_GAINED,SPLICE_SITE,STOP_LOST,SPLICE_SITE,FRAMESHIFT_CODING,SPLICE_SITE,NON_SYNONYMOUS_CODING,SPLICE_SITE,SYNONYMOUS_CODING,SPLICE_SITE,SYNONYMOUS_CODING,5PRIME_UTR,SPLICE_SITE,5PRIME_UTR,3PRIME_UTR,SPLICE_SITE,3PRIME_UTR,INTRONIC,ESSENTIAL_SPLICE_SITE,INTRONIC,SPLICE_SITE,INTRONIC,UPSTREAM,DOWNSTREAM]"*

> entrez = getBM("entrezgene",filters=c("chromosome_name","snptype_filters"), values=list(22,"NON_SYNONYMOUS_CODING"),mart=ensembl)

> entrez[1:5,]

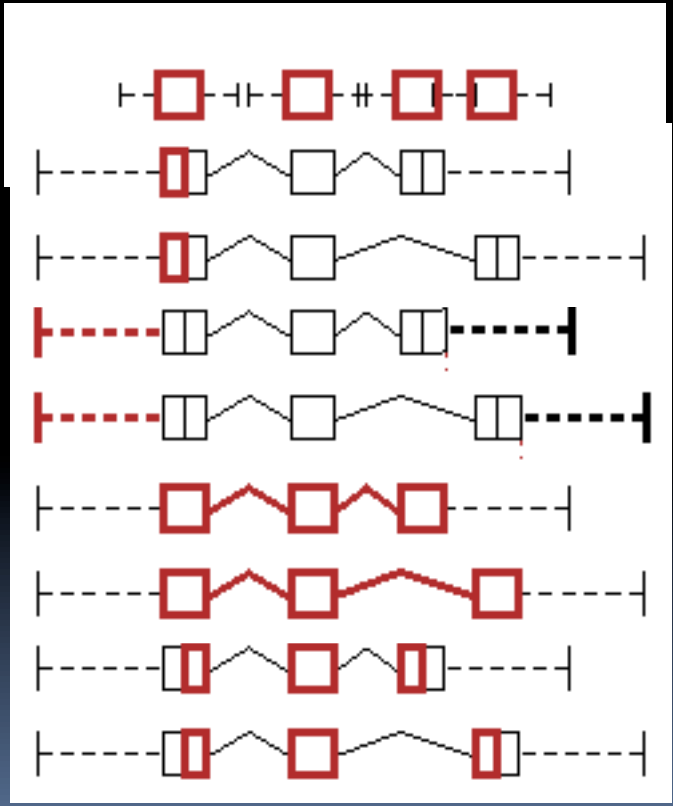
> [1] 23784 81061 150160 150165 128954

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
    TACAAGGGTCAGGTGCCTGAGAACGAGGCTAACGTCGTAATCACCACACTGA
    AAGTGACTGATGCTGATGCCCCCAATACCCCAGCGTGGGAGGCTGTATACACC
    ATATTGAATGATGATGGTGGACAATTTGTCGTCACCACAAATCCAGTGAACAA
    CGATGGCATTTTGAAAACAGCAAAG
 hgnc_symbol
1     CDH1
```

TASK 6 - Ensembl

Retrieve 2000bp sequence upstream of the APC and CUL1 translation start site

TASK 6 - Ensembl

```
>promoter=getSequence(id=c
   ("APC","CUL1"),type="hgnc_symbol",
   seqType="coding_gene_flank",upstream =2000,
   mart=ensembl)
```

```
> promoter

    coding_gene_flank
1 TTGTTCATCTGAAGAGTTGATTTTTTTATTCCTGTAATA.....................
2 TCCGTAGCAGTTGAATGTG .....................

hgnc_symbol
1      APC
2      CUL1
```

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: 1434_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
>out = getLDS(attributes=c("affy_hg_u95av2","hgnc_symbol"),
    filters="affy_hg_u95av2", values=c("1888_s_at","1434_at"),mart=human,
    attributesL="affy_chicken", martL=chicken)
> out
      V1  V2            V3
1  1434_at PTEN GgaAffx.25913.1.S1_a
2 1888_s_at  KIT    Gga.606.1.S1_at
```

Ensembl Archives

- Provide alternate host

```
>listMarts(host="may2009.archive.ensembl.org/biomart/martservice/")
                biomart                                  version
1 ENSEMBL_MART_ENSEMBL       Ensembl 54
2 ENSEMBL_MART_SNP           Ensembl Variation 54
3 ENSEMBL_MART_VEGA                  Vega 35
4 REACTOME                           Reactome(CSHL US)
5 wormbase_current                   WormBase (CSHL US)
6 pride                              PRIDE (EBI UK)

>ensembl54=useMart("ENSEMBL_MART_ENSEMBL",
    host="may2009.archive.ensembl.org/biomart/martservice/")
```

Ensembl Archives

- Access to archives by setting archive=TRUE or connect to specific host (Note that this is currently not up to date in the central repository)

```
>listMarts(archive=TRUE)
                biomart                    version
1      ensembl_mart_51          Ensembl 51
2      snp_mart_51              SNP 51
3      vega_mart_51             Vega 32
4      ensembl_mart_50          Ensembl 50
5     snp_mart_50               SNP 50
```

```
> ensembl51 = useMart("ensembl_mart_51", archive=TRUE,
      dataset="hsapiens_gene_ensembl")
```

Variation BioMart

- dbSNP mapped to Ensembl

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

TASK 8 - Variation

Retrieve all refsnp_ids and their alleles and position that are located on chromosome 8 and between bp 148350 and 158612.

TASK 8 - Variation

```
>out=getBM(attributes=c("refsnp_id","allele","chrom_start"),
   filters=c("chr_name","chrom_start","chrom_end"), values=list
   (8,148350, 158612), mart=snp)
> out[1:5,]
     refsnp_id         allele     chrom_start
1 ENSSNP4490669    C/G    148729
2 ENSSNP5558526    T/C    148909
3 ENSSNP4089737    T/A    149060
4 ENSSNP9060169    C/T    149245
5 ENSSNP4351891    C/G    149250
```

HapMap

- public resource that will help researchers find genes associated with human disease and response to pharmaceuticals

- Task 9:

  Retrieve the alleles and allele frequencies of all non-synonymous coding SNPs on chromosome 19 in the Yoruban population

HapMap

```
> hapmap = useMart("HapMap_rel27",
  dataset="hm27_variation_yri")
> yri = getBM(c
  ("chrom","start","alleles","ref_allele","ref_allel
  e_freq","other_allele_freq"),filters=c
  ("chrom","coding_nonsynon"),values=list
  ("chr19",TRUE),mart=hapmap)
```

HapMap

```
> head(yri)
  chrom  start alleles ref_allele ref_allele_freq
    other_allele_freq
1 chr19 244828    C/G          G          0.458
    0.542
2 chr19 244913    C/T          C          0.721
    0.279
3 chr19 244934    A/G          A          0.429
    0.571
4 chr19 278923    C/T          C          0.996
    0.004
5 chr19 285441    C/T          C          1.000
    0.000
6 chr19 313283    C/T          C          0.947
    0.053
```

COSMIC

- Catalogue Of Somatic Mutations In Cancer (Sanger)

Experiments	2760220
Tumours	541928
Mutations	136326
References	10383
Genes	18490
Fusions	4946
Structural Variants	2307
Whole Cancer Genomes	28

Note:
Need devel
version of biomaRt
(>= 2.5.1)

TASK 10 - COSMIC

Retrieve all known mutations in the following two cell lines : MCF7 and BT474

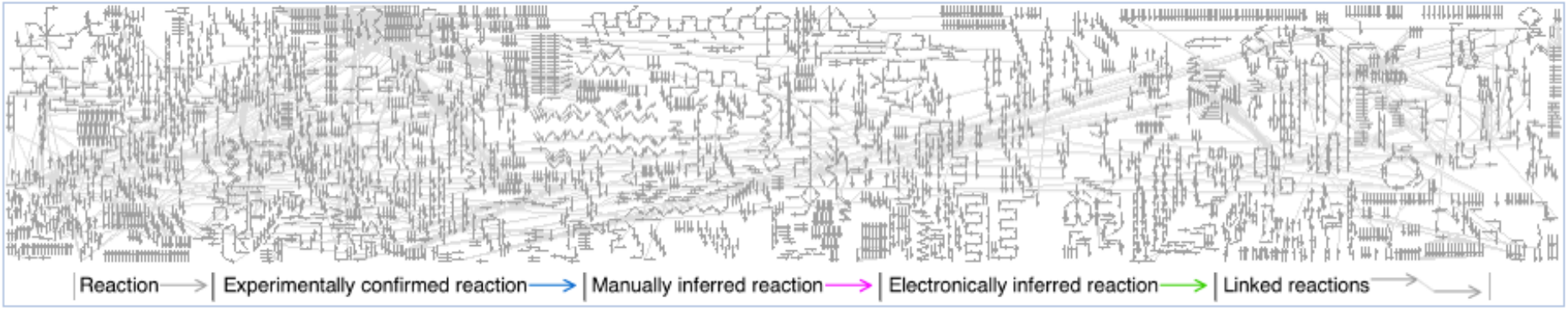
Attributes to query:

sample_name
gene_name
aa_mut_syntax
mut_type_cds
zygosity

TASK 10- COSMIC

```
> cosmic=useMart("CosmicMart",dataset="COSMIC47")
> mut = getBM(c
    ("sample_name","gene_name","aa_mut_syntax","aa_mut_start","mut_type_cds","zygosity"),filt
    ers="sample_name",values=c("MCF7","BT474"),mart=cosmic)
> mut[1:10,]
   sample_name gene_name aa_mut_syntax mut_type_cds   zygosity
1         MCF7    ERBB2                    Unknown
2         MCF7   CDKN2A       p.0?    Deletion Homozygous
3         MCF7     RB1                  Unknown
4         MCF7     CDH1                   Unknown
5         MCF7    STK11                   Unknown
6         MCF7    FBXW7                    Unknown
7         MCF7   PIK3CA     p.E545K    Unknown   Unknown
8         MCF7    SMAD4                    Unknown
9         MCF7     BRAF                  Unknown
10        MCF7     EGFR                   Unknown
```

Reactome



- Reactome is an open-source and manually curated pathway database that provides pathway analysis tools for life science researcher
- http://www.reactome.org
- Task 11:

  Retrieve uniprot ids for human genes involved in the following pathways: DNA Repair, Signaling by WNT,Muscle contraction

Reactome

```
> reactome = useMart("REACTOME",
  dataset="pathway")
> ids = getBM(c
  ("referencedatabase_uniprot","_displayname"),filters=c
  ("_displayname","species_selection"),value=list(c("DNA
  Repair","Signaling by WNT","Muscle contraction"),"Homo
  sapiens"), mart=reactome)
```

Reactome

```
> head(ids)
  referencedatabase_uniprot _displayname
1                    P62988   DNA Repair
2                    P52435   DNA Repair
3                    P36954   DNA Repair
4                    P30876   DNA Repair
5                    O15514   DNA Repair
6                    P62487   DNA Repair
```

Gramene

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

TASK 12 - Gramene

Retrieve the ensembl gene id, external gene id, a description and the start positions of all genes from *Arabidopsis thaliana* that are located on chromosome 1 between basepair 30.000 and 41.000

TASK 12 - Gramene

>gramene = useMart("ENSEMBL_MART_ENSEMBL", dataset=" athaliana_eg_gene*")*

>getBM(c
("ensembl_gene_id","external_gene_id","description","start_position","end_position"), filters=c("chromosome_name","start","end"), values=list("1", "30000","41000"), mart=gramene)

TASK 12 - Gramene

```
> getBM(c("ensembl_gene_id","external_gene_id","description","start_position","end_position"), filters=c("chromosome_name","start","end"), values=list("1", "30000","41000"), mart=gramene)
  ensembl_gene_id external_gene_id
1 AT1G01050-TAIR-G         AtPPa1
2 AT1G01070-TAIR-G      AT1G01070
3 AT1G01040-TAIR-G           DCL1
4 AT1G01060-TAIR-G            LHY

        description
1 AtPPa1 (Arabidopsis thaliana pyrophosphorylase 1); inorganic diphosphatase; Encodes a soluble protein with inorganic pyrophosphatase activity that is highly specific for Mg-inorganic pyrophosphate.
2 nodulin MtN21 family protein; nodulin MtN21 family protein; LOCATED IN: membrane; EXPRESSED IN: 17 plant structures; EXPRESSED DURING: 7 growth stages; CONTAINS InterPro DOMAIN/s: Protein of unknown function DUF6, transmembrane (InterPro:IPR000620); BEST Arabidopsis thaliana protein match is: nodulin MtN21 family protein (TAIR:AT1G11460.1); Has 1705 Blast hits to 1692 proteins in 315 species: Archae - 18; Bacteria - 780; Metazoa - 4; Fungi - 6; Plants - 641; Viruses - 0; Other Eukaryotes - 256 (source: NCBI BLink).
3 DCL1 (DICER-LIKE 1); ATP-dependent helicase/ double-stranded RNA binding / protein binding / ribonuclease III; Encodes a Dicer homolog. Dicer is a RNA helicase involved in microRNA processing. Mutations in this locus can result in embryo lethality. Embryo shape at seed maturity is globular-elongate. Other mutants convert the floral meristems to an indeterminate state, others yet show defects in ovule development. mRNA is expressed in all shoot tissues. DCL1 is able to produce miRNAs and siRNAs.
4 LHY (LATE ELONGATED HYPOCOTYL); DNA binding / transcription factor; LHY encodes a myb-related putative transcription factor involved in circadian rhythm along with another myb transcription factor CCA1

  start_position end_position
1     31170     33153
2     38752     40944
3     23146     31227
```

Wormbase

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

TASK 13 - Wormbase

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

TASK 13 - Wormbase

```
> worm = useMart("wormbase195",
                 dataset="wormbase_rnai")

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

TASK 13 - 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
11 WBRNAi00026915            embryonic_lethal
12 WBRNAi00026915              larval_lethal
```

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
- Sometimes server can be down

Reporting bugs

- Check if http://www.biomart.org is online
- Check with MartView if you get the same output
  - Yes: contact database e.g.

    helpdesk@ensembl.org

  - No: contact me

    sdurinck@lbl.gov

Acknowledgements

- EBI
  - BioMart Team
    - Arek Kasprzyk
    - Syed Haider
  - Ensembl Team
    - Rhoda Kinsella
    - Ewan Birney
- EMBL
  - Wolfgang Huber

Bioconductor users