Bio conductor Jan 2011 Overview course: Prosp ectus

© VJ Carey
December 23, 2010

Contents

1 Introduction

This course aims to provide a broad overview of Bioconductor facilities and approaches to analysis of genome scale data. The methods used are very concrete. We suppose that there is general interest in exible computing with all of the following:

> Consensus genomic sequence (e.g., 3 billion ordered nucleotides organized in 24 chromosomes for H. sapiens);
>
> Nomenclatures and characteristics of known genes (e.g., names and aliases, chromosomal and cytoband addresses, TSS and strand);
>
> Reference data structures of computational and systems biology (e.g., the archive of abstracts maintained at National Library of Medicine PubMed, the Gene Ontology directed acyclic graph and term de nitions, the Kyoto Encyclopedia of Genes and Genomes pathway library);
>
> Structural metadata about widely used genome-scale assays (e.g., characterization of commercial microarray probe sequences and chip geometries);
>
> Published genome-scale experiments, as archived in NCBI Gene Expression Omnibus (GEO), EBI ArrayExpress, NCBI Short Read Archive (SRA), the 1000 genomes (1KG) project, and other institutional or privately de ned archives;
>
> Reliable analytical workows that transform locally generated or publicly archived experimental data resources into resources for inference, interpretation, and, in many cases, for extension or elaboration.

These are high-level biocomputing targets. Many prospective students have speci c experiments or even les that they need to interpret. This course may or may not provide direct solutions to such problems. The intention of the course is not to work through one-o solutions but to exhibit { in a concrete way { reusable resources and principles of analysis that can be specialized as needed.

The one-day course plan (subject to change) is: 0830-0930 R in brief; bioconductor installation and web resources

0930-1030 Annotation maps and gene sets; Entrez, GO, KEGG, msigDB

1030-1100 break

1100-1145 GenomicFeatures and GenomicRanges

1145-1245 Affy and Illumina expression arrays: preprocessing and ExpressionSet construction; GEOquery; GEOmetadb

1245-1330 lunch on your own

1330-1500 Differential gene expression, gene set analysis

1500-1530 break

1530-1615 Genotyping arrays; GWAS representation and analysis

1615-1700 Sec-gen sequencing intro: import, QA and basic differential expression with RNA-seq

2 Language, computational models

A tutorial document on R will be made available to all registrants upon registration. Briey, students must be aware that R is an interpreted language, that will be used interactively. Data can be imported for programming in various ways, and a few common approaches will be exhibited. All computations in R proceed by evaluation of functions. Functions operate on values to produce new values. Values can have various structures, ranging from individual numbers or strings to hiearchically de ned combinations of basic data structures.

From the perspective of data analysis, a very common situation occurs when we have a two-dimensional array or table of values where rows correspond to objects of interest and columns correspond to properties measured on the objects. If the table is the value x, then the R expression f(x) generates some new value depending on the code de ning f. Students should be able to think of values x and functions f that represent interesting operations prior to coming to the course. It will be useful if the students have already composed some of their own functions.

R courses spend considerable time on the content of the previous two paragraphs. This Bioconductor course takes much of that for granted and aims to promote expertise in thinking broadly about resources for and methods in computational biology. For example, we want to be able to survey the Gene Expression Omnibus data resource for microarray archives related to diseases or biological processes that we specify. In the class we will learn how to use the GEOmetadb package and R programming to de ne the function qmeta that is used as follows:

```
> rs = qmeta(title = "asthma", organism = "Homo sapiens", con = con) Some information from rs is > dim(rs)
[1] 12 2 >
rs[5,]
                                                                                    title 5
Expression in induced sputum during acute exacerbations in
    asthmatic children with/without chronic airflow obstruction gse
```

5 GSE19903 showing that 12 GEO series have ’asthma’ in their title, and showing the level of detail

achieved in some titles. To get this result we have performed, in R, computations on a SQLite database

to nd information about experiments in GEO. The qmeta function returns a tabular structure called a data.frame, and we extracted one line of the data frame to obtain the title and label for the experiment on asthma exacerbations.

This concludes the prospectus commentary on the language and computational models of interest in the overview course. We have endeavored to show how R's exibility and interoperability can be leveraged to acquire information that can satisfy general curiosities about genome-scale experimental data available in curated archives. It is natural to move from this (programmatically) small-scale exercise to use of R's pattern-matching and more general text mining facilities (Feinerer et al., 2008). Below we will examine the results of acquiring the associated processed array data using the GEOquery package (Davis and Meltzer, 2007).

3 Annotation of organisms, functions, and platforms

3.1 Institutionally de ned metadata maps

The NCBI-based Entrez Gene catalog of \genes" is systematically harvested and packaged for use with Bioconductor. In the following we generate a report on mappings from Entrez identi ers to various metadata elements, followed by a series of examples.

> library(org.Hs.eg.db) >
org.Hs.eg()

Quality control information for org.Hs.eg:

This package has the following mappings:

org.Hs.egACCNUM has 30045 mapped keys (of 44811 keys)
org.Hs.egACCNUM2EG has 656242 mapped keys (of 656242 keys)
org.Hs.egALIAS2EG has 110538 mapped keys (of 110538 keys)
org.Hs.egCHR has 44424 mapped keys (of 44811 keys)
org.Hs.egCHRLENGTHS has 93 mapped keys (of 93 keys)
org.Hs.egCHRLOC has 22107 mapped keys (of 44811 keys)
org.Hs.egCHRLOCEND has 22107 mapped keys (of 44811 keys)
org.Hs.egENSEMBL has 19496 mapped keys (of 44811 keys)
org.Hs.egENSEMBL2EG has 19887 mapped keys (of 19887 keys)
org.Hs.egENSEMBLPROT has 19461 mapped keys (of 44811 keys)
org.Hs.egENSEMBLPROT2EG has 75463 mapped keys (of 75463 keys)
org.Hs.egENSEMBLTRANS has 19494 mapped keys (of 44811 keys)
org.Hs.egENSEMBLTRANS2EG has 109368 mapped keys (of 109368 keys)
org.Hs.egENZYME has 2142 mapped keys (of 44811 keys)
org.Hs.egENZYME2EG has 936 mapped keys (of 936 keys)
org.Hs.egGENENAME has 44811 mapped keys (of 44811 keys) org.Hs.egGO has 17794 mapped keys (of 44811 keys) org.Hs.egGO2ALLEGS has 13360 mapped keys (of 13360 keys) org.Hs.egGO2EG has 10161 mapped keys (of 10161 keys) org.Hs.egMAP has 37845 mapped keys (of 44811 keys) org.Hs.egMAP2EG has 2601 mapped keys (of 2601 keys) org.Hs.egOMIM has 14704 mapped keys (of 44811 keys) org.Hs.egOMIM2EG has 17368 mapped keys (of 17368 keys) org.Hs.egPATH has 5501 mapped keys (of 44811 keys) org.Hs.egPATH2EG has 214 mapped keys (of 214 keys) org.Hs.egPFAM has 24976 mapped keys (of 44811 keys) org.Hs.egPMID has 30298 mapped keys (of 44811 keys)

org.Hs.egPMID2EG has 283543 mapped keys (of 283543 keys)
org.Hs.egPROSITE has 24976 mapped keys (of 44811 keys)
org.Hs.egREFSEQ has 28641 mapped keys (of 44811 keys)
org.Hs.egREFSEQ2EG has 91755 mapped keys (of 91755 keys)
org.Hs.egSYMBOL has 44811 mapped keys (of 44811 keys)
org.Hs.egSYMBOL2EG has 44796 mapped keys (of 44796 keys)
org.Hs.egUCSCKG has 20528 mapped keys (of 44811 keys)
org.Hs.egUNIGENE has 25212 mapped keys (of 44811 keys)
org.Hs.egUNIGENE2EG has 25814 mapped keys (of 25814 keys)
org.Hs.egUNIPROT has 18990 mapped keys (of 44811 keys)

Additional Information about this package:

DB schema: HUMAN_DB DB schema version:
2.1 Organism: Homo sapiens Date for NCBI
data: 2010-Sep7 Date for GO data: 20100904
Date for KEGG data: 2010-Sep7 Date for
Golden Path data: 2010-Mar22 Date for IPI
data: 2010-Aug19 Date for Ensembl data:
2010-Aug5

```
> get("1000", org.Hs.egCHRLOC)

           18
-25530929
```

```
> get("1000", org.Hs.egSYMBOL)

[1] "CDH2"
```

```
> gomap = get("1000", org.Hs.egGO) >
length(gomap)

[1] 29
```

```
> names(gomap)[1:4]

[1] "GO:0007157" "GO:0007155" "GO:0007156" "GO:0007416"
```

```
> keggmap = get("1000", org.Hs.egPATH)
7
```

3.2 Maps of functions and pathways

Gene Ontology is a consortium-de ned vocabulary about genes and gene products and their functions, organized into an ontology. This is also packaged for use with Bioconductor; we continue the previous exercise:

> library(GO.db) > sapply(names(gomap)[1:4], function(x) Term(get(x, GOTERM)))

GO:0007157 GO:0007155 "heterophilic cell-cell adhesion" "cell adhesion"

GO:0007156 GO:0007416 "homophilic cell adhesion" "synapse assembly"

KEGG is the Kyoto Encyclopedia of Genes and Genomes. A somewhat di erent interrogation pattern is used:

> library(KEGG.db) > mget(keggmap, KEGGPATHID2NAME)

$ 04514 [1] "Cell adhesion molecules (CAMs)"

$ 05412 [1] "Arrhythmogenic right ventricular cardiomyopathy (ARVC)"

It would be good for the student to have a clear understanding of why these interrogations have such di erent forms (mget vs sapply). What happens when mget is used with the GO query?

3.3 Annotations for A ymetrix(tm) arrays

3.3.1 Pre 1.0 ST series expression arrays > library(hgu133plus2.db)

> hgu133plus2()

Quality control information for hgu133plus2:

This package has the following mappings:

hgu133plus2ACCNUM has 54675 mapped keys (of 54675 keys)
hgu133plus2ALIAS2PROBE has 73685 mapped keys (of 110538 keys)
hgu133plus2CHR has 40772 mapped keys (of 54675 keys)

hgu133plus2CHRLENGTHS has 93 mapped keys (of 93 keys)
hgu133plus2CHRLOC has 39212 mapped keys (of 54675 keys)
hgu133plus2CHRLOCEND has 39212 mapped keys (of 54675 keys)
hgu133plus2ENSEMBL has 37294 mapped keys (of 54675 keys)
hgu133plus2ENSEMBL2PROBE has 18042 mapped keys (of 19887 keys)
hgu133plus2ENTREZID has 40801 mapped keys (of 54675 keys)
hgu133plus2ENZYME has 4616 mapped keys (of 54675 keys)
hgu133plus2ENZYME2PROBE has 928 mapped keys (of 936 keys)
hgu133plus2GENENAME has 40801 mapped keys (of 54675 keys)
hgu133plus2GO has 35250 mapped keys (of 54675 keys)
hgu133plus2GO2ALLPROBES has 13288 mapped keys (of 13360 keys)
hgu133plus2GO2PROBE has 10091 mapped keys (of 10161 keys)
hgu133plus2MAP has 40571 mapped keys (of 54675 keys)
hgu133plus2OMIM has 27795 mapped keys (of 54675 keys)
hgu133plus2PATH has 11297 mapped keys (of 54675 keys)
hgu133plus2PATH2PROBE has 214 mapped keys (of 214 keys)
hgu133plus2PFAM has 39581 mapped keys (of 54675 keys)
hgu133plus2PMID has 40160 mapped keys (of 54675 keys)
hgu133plus2PMID2PROBE has 276342 mapped keys (of 283543 keys)
hgu133plus2PROSITE has 39581 mapped keys (of 54675 keys)
hgu133plus2REFSEQ has 40248 mapped keys (of 54675 keys)
hgu133plus2SYMBOL has 40801 mapped keys (of 54675 keys)
hgu133plus2UNIGENE has 40632 mapped keys (of 54675 keys)
hgu133plus2UNIPROT has 37123 mapped keys (of 54675 keys)

Additional Information about this package:

DB schema: HUMANCHIP_DB DB schema version: 2.1 Organism: Homo sapiens Date for NCBI data: 2010-Sep7 Date for GO data: 20100904 Date for KEGG data: 2010-Sep7 Date for Golden Path data: 2010-Mar22 Date for IPI data: 2010-Aug19 Date for Ensembl data: 2010-Aug5

Quality control information for hgu133plus2:

This package has the following mappings:
9

hgu133plus2ACCNUM has 54675 mapped keys (of 54675 keys)
hgu133plus2ALIAS2PROBE has 73685 mapped keys (of 110538 keys)
hgu133plus2CHR has 40772 mapped keys (of 54675 keys)
hgu133plus2CHRLENGTHS has 93 mapped keys (of 93 keys)
hgu133plus2CHRLOC has 39212 mapped keys (of 54675 keys)
hgu133plus2CHRLOCEND has 39212 mapped keys (of 54675 keys) ...

The interface here is similar to that shown for the Entrez map. Keys have the familiar tokenization:

> mappedkeys(hgu133plus2CHRLOC)[1:5] [1] "1007_s_at" "1053_at" "117_at" "121_at" "1255_g_at"

3.3.2 Probe sequences and mappings for the 1.0 ST series There are various approaches to working with A ymetrix arrays with R. I will consider approaches related to the oligo package (Carvalho and Irizarry, 2010) and to the a y package (Gautier et al., 2004).

The GSE series on asthma noted above is based on the a y gene 1.0 ST platform. We can learn about this platform in various ways. For example, we can look at all the probe sequences on the array; here are ve.

> library(hugene10stv1probe) > as.data.frame(hugene10stv1probe[1000:1005,])

sequence x y Probe.Set.Name 1000 GGTCGCTGGGCTGCTCGGGGGCCAC 1031 1047 1100382 1001 TCCGGAGGTCGCTGGGCTGCTCGGG 923 750 788424 1002 CTATCCGGAGGTCGCTGGGCTGCTC 340 766 804641 1003 AGGCCGGGCGCGCAGGGCCGGGCCG 1047 706 742348 1004 GGGCTGAGAGGCCGGGCGCGCAGGG 1034 879 923985 1005 TCCGTGTGGTTGATCGGGTAGAAGC 182 451 473733

Probe.Interrogation.Position Target.Strandedness 1000 1568466 Sense 1001 1568472 Sense 1002 1568475 Sense 1003 1568520 Sense 1004 1568528 Sense 1005 1568567 Sense

It is somewhat sobering to nd that most of these sequences are not uniquely situated in the genome. We will indicate how to check this programmatically.

Note that the probe set identi ers are numeric in form, utterly semantically opaque. We have another annotation mapping package:

> library(hugene10stprobeset.db) > ck =
mappedkeys(hugene10stprobesetCHR) > ck[1:5]

[1] "7896741" "7896743" "7896745" "7896755" "7896757" > intersect(ck[1:5],
as.data.frame(hugene10stv1probe)[, 4]) character(0) There's no connection. But with the oligo-oriented platform description package (built
with pdInfoBuilder) we can nd: >
library(pd.hugene.1.0.st.v1)
> con = pd.hugene.1.0.st.v1@getdb() > con

<SQLiteConnection: DBI CON (636, 10)> > dbListTables(con) [1] "chrom_dict"
"core_mps" "featureSet" "level_dict" "pmfeature"
[6] "table_info" "type_dict" > dbGetQuery(con, "select * from
pmfeature limit 5")

fid fsetid atom x y 1 116371
7892501 1 870 110 2 943979 7892501
2 28 899 3 493089 7892501 3 638 469
4 907039 7892501 5 888 863 5
1033309 7892502 7 108 984

and we see that the probe package is indexed using feature identi ers (d) while the probeset.db package is indexed using feature set identi ers (fsetid). This is what allows us to map from 25mers to asserted genomic locations and gene assignments.

3.4 Annotation of Illumina expression arrays

There have been many versions of nomenclatures for Illumina probes. For example
> library(illuminaHumanv1.db)
> mappedkeys(illuminaHumanv1CHR)[1:5] [1] "GI_10047089-S"
"GI_10047091-S" "GI_10047093-S" "GI_10047099-S"
[5] "GI_10047103-S"

The lumi developers at Northwestern U. noted that the 50-mer can be base-64 encoded to a more convenient string that can be mapped without regard to platform version (Du et al., 2008).

```
> library(lumiHumanAll.db) > mk =
mappedkeys(lumiHumanAllCHR)[1:5] > mk

[1] "0..gjSdbqnr.rUSTBI" "0..iEt2d7VUp6LkunY" "0.1KICpXRq4mqYep1U" [4]
"0.3Uzozz.u0qpThYsk" "0.D0iq3wJKnqWirurU"

> library(lumi) >
id2seq(mk)

                                                0..gjSdbqnr.rUSTBI
"TTTTTTGAAGATCAGCTCCGTGGGGCTGGTTTTGGTCCAC
AGCATAACAG"
                                                0..iEt2d7VUp6LkunY
"TTTTTTGAGACAGTCTCGCTCTGTCCCCCAGGCTGGAGTG
CAGTGGCTCG"
                                                0.1KICpXRq4mqYep1U
"TTTTCCAGGAGAAAGGGCCCTCACGGGTGAGCGGGGCGA
CTGGGCTCCCC"
                                                0.3Uzozz.u0qpThYsk
"TTTTCTCCATATGGATATTATTTTGTGTCAGGGGGCCATGA
CCGAGTAGC"
                                                0.D0iq3wJKnqWirurU
"TTTAATTCAGAGGGGTCTTAAAGCAGGGCTGGGCCGGAG
GGTGTGGGTCC"
```

3.5 Full genomic sequence and transcript databases

Various builds of genomic sequences for a number of organisms are available in convenient forms.

```
> library(BSgenome.Hsapiens.UCSC.hg19) > c1
= Hsapiens$chr1 > c1

   249250621-letter "MaskedDNAString" instance (# for masking) seq:
########################################...########################################
masks:
   maskedwidth maskedratio active names desc 1 23970000 0.09616827 TRUE AGAPS
assembly gaps 2 0 0.00000000 TRUE AMB intra-contig ambiguities (empty) 3
114014472 0.45742904 FALSE RM RepeatMasker 4 1581889 0.00634658 FALSE TRF
```

Tandem Repeats Finder [period<=12] all masks together:
12

maskedwidth maskedratio
138071094 0.5539448

all active masks together:
maskedwidth maskedratio

23970000 0.09616827 With this rich information source in hand, let’s consider how to create a piece of

software that veri es assertions about probe sequences on the human gene 1.0 ST chip (v1). For convenience, we will allow the user to provide a gene symbol, and our function will determine the sequences interrogating the gene’s transcripts and check that they reside on the chromosome asserted in the probeset annotation package.

```
> check1.0 = function(sym, genomepkg = "BSgenome.Hsapiens.UCSC.hg19", + userevcomp
= TRUE) { + require(genomepkg, character.only = TRUE) + require(pd.hugene.1.0.st.v1) +
require(hugene10stprobeset.db) + require(hugene10stv1probe) + fsetid = get(sym,
revmap(hugene10stprobesetSYMBOL)) + if (length(fsetid) == 1 && is.na(fsetid)) + stop("could
not find symbol in reverse map") + chr = unique(paste("chr", unlist(mget(fsetid,
hugene10stprobesetCHR)), + sep = "")) + st = sapply(mget(fsetid,
hugene10stprobesetCHRLOC), "[", + 1) + en = sapply(mget(fsetid,
hugene10stprobesetCHRLOCEND), "[", + 1) + if (length(chr) > 1) + stop("more than one
chromosome identified") + mapcon = pd.hugene.1.0.st.v1@getdb() +
on.exit(dbDisconnect(mapcon)) + inclause = paste("(", paste(sQuote(fsetid), collapse = ","), +
")") + fidquery = paste("select fid from pmfeature where fsetid in", + inclause, sep = "") + fid =
dbGetQuery(mapcon, fidquery)[[1]] + if (length(fid) < 1) + stop("no fids found") + mfid =
match(fid, hugene10stv1probe[, "Probe.Set.Name"]) + if (length(mfid) < 1) + stop("could not
find probe sequences") + seqs = DNAStringSet(strs <- hugene10stv1probe[mfid,
"sequence"]) + if (userevcomp)
```

```
+ seqs = reverseComplement(seqs) + mind = matchPDict(PDict(seqs),
Hsapiens[[chr]]) + nhits = countIndex(mind) + if (max(nhits) == 0) + stop("there
were no hits; check userevcomp setting") + firstst = sapply(startIndex(mind), "[",
1) + firsten = sapply(endIndex(mind), "[", 1) + data.frame(sym = sym, chr = chr,
seqs = strs, nhits = nhits, + start1 = firstst, end1 = firsten, assertstart = st[1], +
assertend = en[1]) + } > if (!exists("cbr")) cbr = check1.0("CXCL10", userevcomp =
FALSE) > dim(cbr)

[1] 28 8 > cbr[1:5,
]

      sym chr seqs nhits start1 end1 assertstart 1 CXCL10 chr4
GGAATTGTATGTAGGTAGCCACTGA 1 76942595 76942619 -76942272 2 CXCL10 chr4
TAGCCACTGAAAGAATTTGGGCCCC 1 76942610 76942634 -76942272 3 CXCL10 chr4
GAAGCAGGGTCAGAACATCCACTAA 1 76942799 76942823 -76942272 4 CXCL10 chr4
TAAGAACATAGCACCTCAGTAGAGC 1 76942821 76942845 -76942272 5 CXCL10 chr4
GCCTCTGTGTGGTCCATCCTTGGAA 1 76943028 76943052 -76942272
  assertend 1
-76944650 2
-76944650 3
-76944650 4
-76944650 5
-76944650
```

Students interested in software development could consider how to improve this function in various ways. First, the handling of strandedness is completely naive; it is possible that some probes map directly, others only after reverse complementation. Second, it would be useful to return an object with more information { for example, the vector of feature set identi ers can't readily be included in the returned data frame for the current design. An S4 object could manage this straightforwardly. Third, provenance information on the results should be conveyed. Fourth, the orientation towards a speci c organism is unnecessary. Many 'hard-coded' references to speci c platforms can be made more general.

To connect these ndings on genomic locations of probe sequences to transcript structure, note that coordinates needed to decompose genes into transcripts and subsequently into exons are also provided based on UCSC or BioMart tables, via the GenomicFeatures package. At this point, nomenclatures again become complicated; we can use RefSeq, UCSC 'known gene', or ensembl vocabularies at a relatively high level.

3.6 Interaction with genome browsers

The rtracklayer package allows programmatic interaction with various browsers, permitting extraction of track-based information or uploading of tracks recording results of analyses (Lawrence et al., 2009).

3.7 Hyperlinked annotation reports

A tiny bit of code like > ks = mappedkeys(hgu133plus2CHRLOC)[1:5]
> aa = aafTableAnn(ks, "hgu133plus2.db") > saveHTML(aa, file="~/aa.html")

can be used to generate a hyperlinked report like

Probe	Symbol	Description	Chromosome	Chromosome Location	GenBank	Gene	Cytoband	UniGene	PubMed	Gene Ontology	Pathway
1007_s_at	DDR1	discoidin domain receptor tyrosine kinase 1	6	30851860, 30852326, 30856464, 2363958, 2364424, 2368562, 2145962, 2146428, 2150566, 2199945, 2200411, 2204549, 2233817, 2234283, 2144847, 2145313, 2149451	U48705	780	6p21.3	Hs.631988	70	nucleotide binding transmembrane receptor protein tyrosine kinase activity receptor activity protein binding ATP binding extracellular region integral to plasma membrane protein amino acid phosphorylation cell adhesion transmembrane receptor protein tyrosine kinase signaling pathway membrane transferase activity	
1053_at	RFC2	replication factor C (activator 1) 2, 40kDa	7	-73645833	M87338	5982	7q11.23	Hs.647062	37	nucleotide binding DNA clamp loader activity protein binding ATP binding nucleus nucleoplasm DNA replication factor C complex DNA replication nucleotide-excision repair, DNA gap filling nucleoside-triphosphatase activity	DNA replication Nucleotide excision repair Mismatch repair
											Spliceosome MAPK signaling

The facility is fairly general, not limited to a y chips, despite the name.

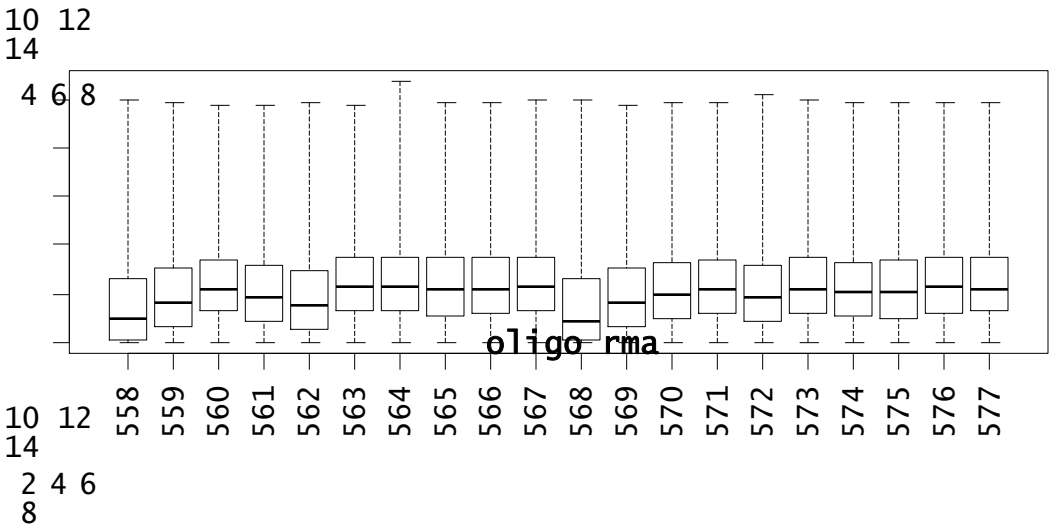
4 Expression microarrays

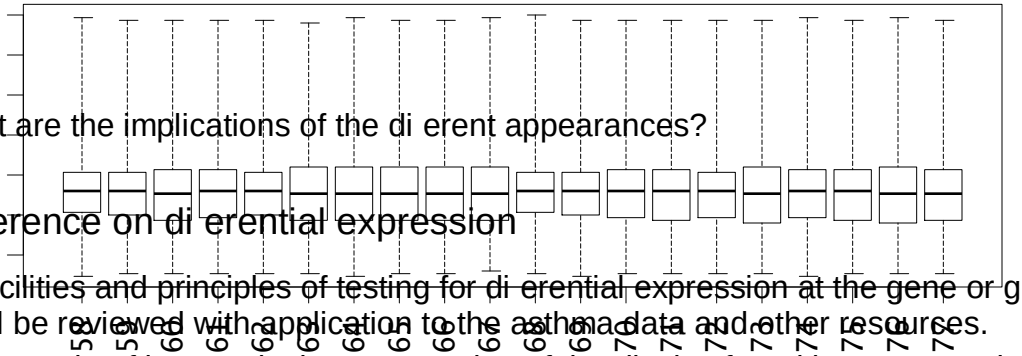
4.1 Mining public resources

We will show how the GEOquery package can be used to acquire a representation of the arrays underlying the asthma exacerbation series identi ed above; direct queries allow us to acquire the original CEL les. Programming to assign arrays to appropriate phenotypic classes, and to encode these characteristics in ExpressionSet instances will be reviewed.

4.2 Preprocessing

We will consider the e ects of di erent choices of preprocessing methods on quanti cation of expression. Spike-in experiments have been useful in helping to level the playing eld for comparisons, and relevant resources will be reviewed. An illustration of a comparison of preprocessing outcomes on a given dataset is:





What are the implications of the di erent appearances?

4.3 Inference on di erential expression

Basic facilities and principles of testing for di erential expression at the gene or gene set level will be reviewed with application to the asthma data and other resources.

An example of interest is the construction of the display found in Bosco et al. (2010):

We will obtain the lists of genes corresponding to some of the di erent modules identi ed in this schematic and examine details of di erential expression between children with better or worse lung function during asthma exacerbation.

18

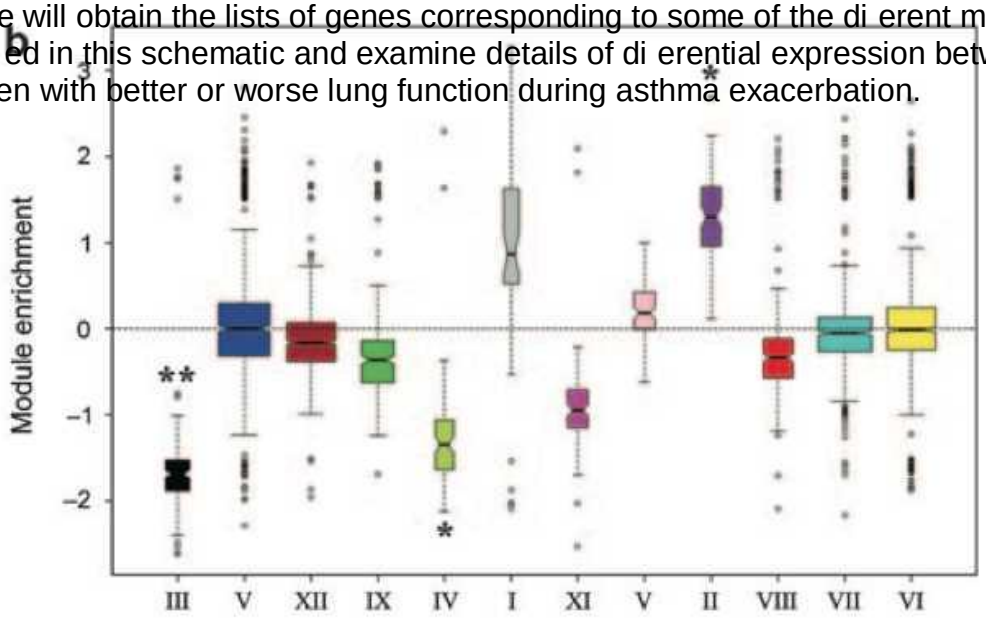
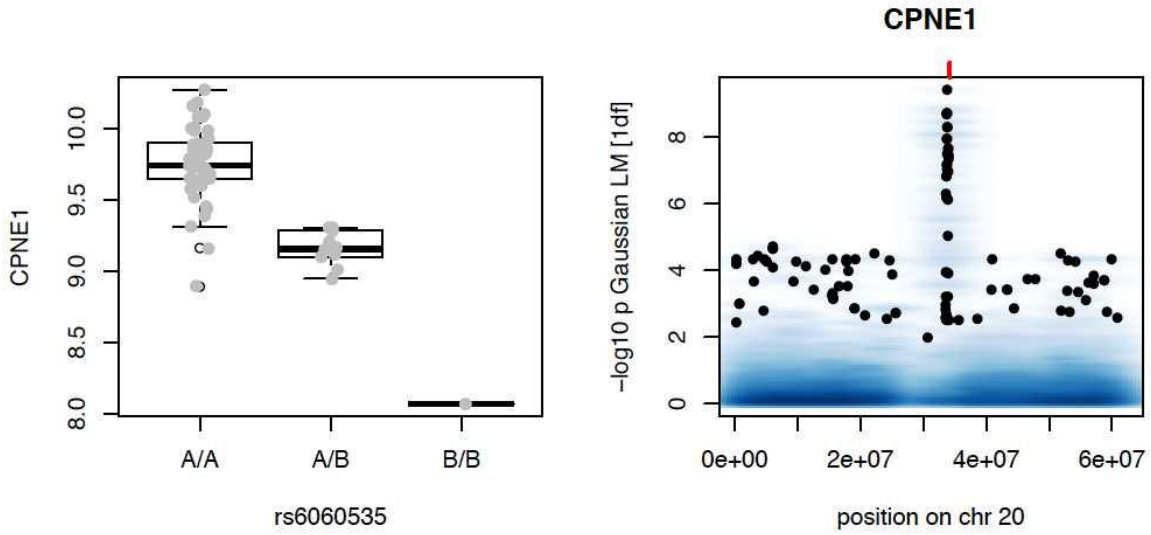


Figure 1 Exacerbation responses are associated with baseline forced expiratory volume in 1 s/forced vital capacity (FEV1/FVC) ratios. Gene expression was profiled by microarray in sputum samples obtained during an acute exacerbation from asthmatic children with (n=10) or without (n=10) deficits in enrollment/baseline FEV1/FVC ratios. (**a**) Modules of coexpressed genes were identified by reverse engineering gene network analysis of the microarray data set (n=20). (**b**) The modules were tested for differential expression in responses from

5 Microarrays for genetics; GWAS

The use of SNP chips to develop cohorts of individuals genotyped at high-resolution is relatively routine. We will review the snpMatrix package and its representation of genotypes (Clayton and Leung, 2007); advanced discussion (requiring R 2.13) will be brief but will consider how snpMatrix2 addresses testing with imputed genotypes and how it can perform its own imputations.

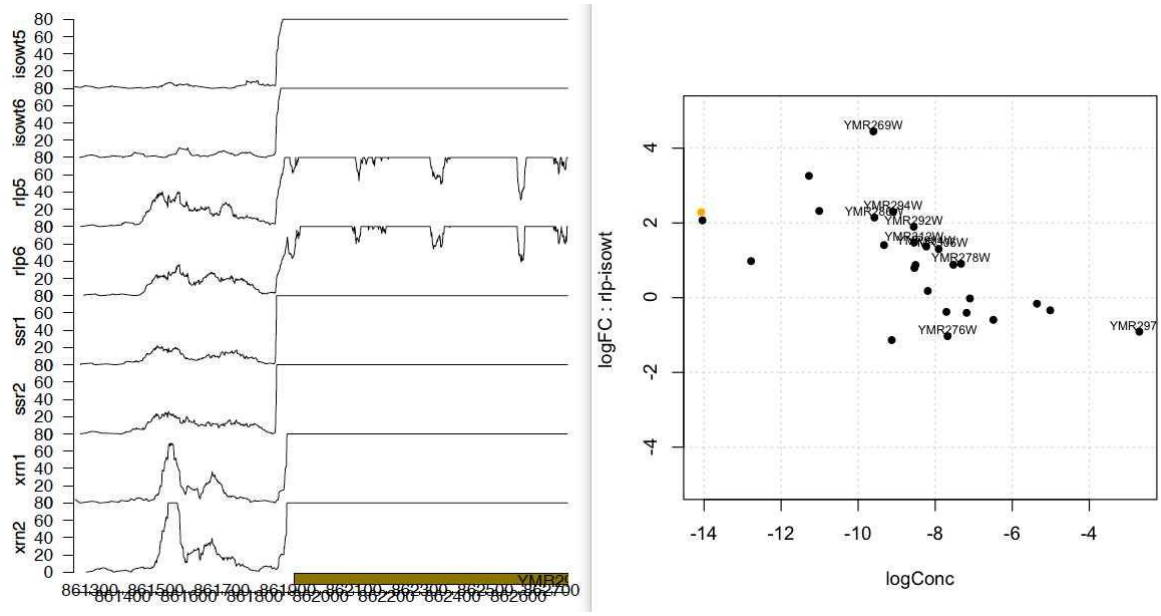
The most basic concerns are encapsulated in displays like the following:



On the left we have a depiction of the association between a phenotype (expression values of a given gene) and allele counts for a given SNP. On the right we have a depiction of a survey of all SNPs on a chromosome for associations with the phenotype. Approaches in R to perform and interpret such surveys e ciently will be discussed (Carey et al., 2009).

6 Short read assays; RNA-seq

For completeness we will conclude with a brief session on sec-gen sequencing. Basic concepts of QA, ltering and tabulation of aligned reads, and tests of di erential expression will be reviewed with data on yeast (Lee et al., 2008).



The displays above are created on the basis of the leeBamViews experimental data package. On the left is a schematic indicating the existence of strain-speci c transcription in an intergenic region of yeast. On the right is a display of test statistics comparing expression between two yeast strains. We will indicate how Rsamtools, GenomeGraphs, edgeR and DEseq can be used for elementary interpretation of RNA-seq experiments.

7 Session information

> sessionInfo()

R version 2.12.0 Patched (2010-11-28 r53696) Platform: x86_64-apple-darwin10.4.0/x86_64 (64-bit)

locale: [1] C

attached base packages: [1] stats graphics grDevices datasets tools utils methods [8] base

other attached packages: [1] annaffy_1.22.0 lumi_2.2.0 [3] BSgenome.Hsapiens.UCSC.hg19_1.3.16 BSgenome_1.18.2 [5] Biostrings_2.18.2 GenomicRanges_1.2.2 [7] IRanges_1.8.7 lumiHumanAll.db_1.10.1 [9] illuminaHumanv1.db_1.8.0 pd.hugene.1.0.st.v1_3.0.2

[11] oligo_1.14.0 oligoClasses_1.12.1 [13] hugene10stprobeset.db_6.0.2 hugene10stv1probe_2.7.0 [15] hgu133plus2.db_2.4.5 KEGG.db_2.4.5 [17] GO.db_2.4.5 org.Hs.eg.db_2.4.6 [19] AnnotationDbi_1.12.0 GEOmetadb_1.10.0 [21] RSQLite_0.9-4 DBI_0.2-5 [23] GEOquery_2.16.3 Biobase_2.10.0 [25] weaver_1.16.0 codetools_0.2-6 [27] digest_0.4.2

loaded via a namespace (and not attached): [1] KernSmooth_2.23-4 MASS_7.3-9 Matrix_0.999375-46 [4] RCurl_1.5-0 XML_3.2-0 affxparser_1.22.0 [7] affy_1.28.0 affyio_1.18.0 annotate_1.28.0

[10] grid_2.12.0 hdrcde_2.15 lattice_0.19-13 [13] methylumi_1.6.1 mgcv_1.7-2 nlme_3.1-97 [16] preprocessCore_1.12.0 splines_2.12.0 xtable_1.5-6

8 References

A Bosco, S Ehteshami, D A Stern, and F D Martinez. Decreased activation of inammatory networks during acute asthma exacerbations is associated with chronic airow obstruction. Mucosal Immunol, 3(4):399{409, Jul 2010. doi: 10.1038/mi.2010.13. URL http://www.nature.com/mi/journal/vaop/ncurrent/full/mi201013a.html.

Vincent J Carey, Adam R Davis, Michael F Lawrence, Robert Gentleman, and Benjamin A Raby. Data structures and algorithms for analysis of genetics of gene expression with bioconductor: Ggtools 3.x. Bioinformatics, 25(11):1447{1448, Jan 2009. doi: 10.1093/bioinformatics/btp169.

Benilton S Carvalho and Rafael A Irizarry. A framework for oligonucleotide microarray preprocessing. Bioinformatics (Oxford, England), 26(19):2363{7, Oct 2010. doi: 10.1093/bioinformatics/btq431.

David Clayton and Hin-Tak Leung. An r package for analysis of whole-genome association studies. Hum Hered, 64(1):45{51, Jan 2007. doi: 10.1159/000101422.

Sean Davis and Paul Meltzer. Geoquery: a bridge between the gene expression omnibus (geo) and bioconductor. Bioinformatics, 14:1846{1847, 2007.

P. Du, W.A. Kibbe, and S.M. Lin. lumi: a pipeline for processing illumina microarray. Bioinformatics, 24(13):1547{1548, 2008.

Ingo Feinerer, Kurt Hornik, and David Meyer. Text mining infrastructure in r. Journal of Statistical Software, 25(5):1{54, 3 2008. ISSN 1548-7660. URL http://www.jstatsoft.org/v25/i05.

Laurent Gautier, Leslie Cope, Benjamin M. Bolstad, and Rafael A. Irizarry. a y| analysis of a ymetrix genechip data at the probe level. Bioinformatics, 20(3):307{315, 2004. ISSN 1367-4803. doi: http://dx.doi.org/10.1093/bioinformatics/btg405.

M Lawrence, R Gentleman, and V Carey. rtracklayer: an r package for interfacing with genome browsers. Bioinformatics, May 2009. doi: 10.1093/bioinformatics/btp328.

Albert Lee, Kasper Daniel Hansen, James Bullard, Sandrine Dudoit, and Gavin Sherlock. Novel low abundance and transient rnas in yeast revealed by tiling microarrays and ultra high-throughput sequencing are not conserved across closely related yeast species. PLoS Genet, 4(12):e1000299, Nov 2008. doi: 10.1371/journal.pgen. 1000299. URL http://www.plosgenetics.org/article/info%253Adoi%252F10.1371%252Fjournal.pgen.1000299.