

Arkas: Raw Reads To Pathway Analyses In Much Less Time

Timothy J. Triche, Jr, Anthony R. Colombo, Harold Pimentel

20 April, 2016

Contents

1	Introduction	1
2	Reads to Quantification to Annotation	1
2.1	Kallisto Installation	2
3	Gene Wise Analysis	2
3.1	The Measure Depends On The Level	2
3.2	Creating The Design Matrix	3
4	Annotate!	3
4.1	Building Annotation libraries	3
5	Annotating Merged KallistoExperiment Containers	10
6	Gene Wise Analysis	12
6.1	Understanding Gene Wise Analysis Output	12
6.2	Understanding Gene Wise Analysis Output	12

1 Introduction

Kallisto is software developed by Nicolas Bray, Harold Pimentel, Pall Melsted, and Lior Pachter (UC Berkeley) that analyzes 30 million unaligned paired-end RNA-Seq reads in less than 5 minutes on a standard laptop computer. Kallisto quantifies transcript abundance from input RNA-Seq reads by using a process, known as pseudoalignment, which identifies the read-transcript compatibility matrix. Arkas is a BioConductor package that extends functions and utilities for RNA-Seq analysis from raw reads to results in minutes.

2 Reads to Quantification to Annotation

Arkas was designed to reduce the programmatic steps required to quantify and annotate multitudes of sample directories. Arkas calls Kallisto to perform on-the-fly transcriptome indexing and quantification recursively for numerous sample directories. For RNA-Seq projects with numerous sequenced samples, Arkas encapsulates expensive preparatory routines. Arkas programmatically orders FASTQ files output from DNA sequencers and inputs a list required by Kallisto for processing multitudes of demultiplexed reads. The Arkas function ‘runKallisto’ recursively indexes transcriptomes and quantifies abundances for any number of samples.

The function ‘mergeKallisto’ merges quantified output into an object of ofsubclass a KallistoExperiment-class, SummarizedExperiment-class. Standard mutators and accessor methods from SummarizedExperiment-methods are preserved in KallistoExperiment-methods. Gene annotation is performed from user-selected bundled transcriptomes (ERCC, Ensembl, and/or RepBase) simultaneously merging annotated samples into one R object: KallistoExperiment. Arkas annotates genes for Homo-Sapiens GrCh38 and Mouse GrCm38 (NCBI). Routines such as ‘annotateBundles’ yields annotated genes from transcriptomes such as External RNA Control Consortium (ERCC), Ensembl release 81 of non-coding RNA, coding RNA, and a hg38 repeatome for both species.

2.1 Kallisto Installation

For linux systems, after installing the dependencies, kallisto is installed via:

```
mkdir /KallistoSource
cd /KallistoSource
git clone https://github.com/pachterlab/kallisto.git
cd ./kallisto
mkdir ./build
cd ./build
cmake ..
make
make install
```

3 Gene Wise Analysis

Arkas supports various levels of analysis, namely transcript-level or gene-level analysis which involves the Limma package for differential expression analysis.

Gene Wise Analysis is founded on the idea that groups of transcripts by a fixed Ensembl Gene ID is termed a “gene”; where “gene” counts are defined as the sum of all transcripts identified by the same unique Ensembl Gene Id. Gene Wise analysis generates bundled and aggregated transcripts associated with a specific Ensembl Gene ID. Arkas wraps limma around another method titled “collapseBundles”, which collapses transcripts into appropriate groups and sums the quantified transcript counts of the group; these transcript aggregated counts are defined as “gene” counts.

3.1 The Measure Depends On The Level

Not all transcripts have the same function homology. Most folks agree that genes are made up by transcripts defined by the transcripts’ coordinate location on the genome. However there are transcript isoforms in DNMT3A and WT1 that have radically different biological function depending on the transcript isoform that is present. The problem with conducting *only* a gene level analysis is that many genes can have the same total gene level total quantified counts; however the biological mechanisms for the same “gene” can vary greatly by a single transcript isoform.

```
suppressWarnings(suppressPackageStartupMessages(library(arkas)))
suppressPackageStartupMessages(library(arkasData))
jsonFile <- system.file("extdata", "NS.JSON", package="arkas")
appSession <- fetchAppSession(jsonFile) ## a
names(appSession$samples) <- appSession$samples ## so column names get set
appSession$outputPath <- system.file("extdata", "", package="arkasData")
pathBase<-system.file("extdata",package="arkasData")
```

```

fastaPath <- paste0(pathBase, "/fasta")
appSession$fastaPath<-fastaPath
NS <- mergeKallisto(appSession$samples,
                    outputPath=appSession$outputPath)

```

3.2 Creating The Design Matrix

In order to analyze bundle-aggregated transcripts defined as “genes”, we create a design matrix which controls for individual effects and contrasts treatment effects across individual subjects.

```

NS$subject <- factor(substr(colnames(NS), 2, 2))
NS$treatment <- substr(colnames(NS), 1, 1) == "s"
NS$ID <- NULL
design <- with(as(colData(NS), "data.frame"),
              model.matrix( ~ treatment + subject ))
rownames(design) <- colnames(NS)
metadata(NS)$design <- design
design

```

```

##      (Intercept) treatmentTRUE subject2 subject4
## n1              1              0          0          0
## n2              1              0          1          0
## n4              1              0          0          1
## s1              1              1          0          0
## s2              1              1          1          0
## s4              1              1          0          1
## attr("assign")
## [1] 0 1 2 2
## attr("contrasts")
## attr("contrasts")$treatment
## [1] "contr.treatment"
##
## attr("contrasts")$subject
## [1] "contr.treatment"

```

4 Annotate!

In order to run gene-wise analysis, Arkas requires that the merged KallistoExperiment must be annotated; this is because we must collapse transcripts into groups linked to unique Ensembl Gene Ids.

4.1 Building Annotation libraries

Library Annotations are built using TxDbLite; these annotation databases allow for lite annotations parsing gene names, bio-types and family type from reference fastas from ERCC, Ensembl, or RepBase. Currently exonic, intronic, or other coordinate dependent information is not included in TxDbLite. The supplemental package arkasData stores the ready-to-load annotation libraries under /extdata/Libraries directory. For demonstration, we build the libraries under the arkasData/extdata/fastq/tmp directory.

```

suppressPackageStartupMessages(library(TxDbLite))
suppressWarnings(suppressPackageStartupMessages(library(arkas)))
suppressPackageStartupMessages(library(arkasData))
jsonFile <- system.file("extdata", "NS.JSON", package="arkas")
appSession <- fetchAppSession(jsonFile)
names(appSession$samples) <- appSession$samples
appSession$outputPath <- system.file("extdata", package="arkasData")
fastaPath<-system.file("extdata", "fasta", package="arkasData")
appSession$fastaPath<-fastaPath
cd<-appSession$fastaPath
setwd(paste0(appSession$fastaPath, "/", "tmp"))
NS <- mergeKallisto(appSession$samples,
                    outputPath=appSession$outputPath)

```

```
## Setting transcriptome automatically from Kallisto call string.
```

```

fastaTx<-c("ERCC.fa.gz", "Homo_sapiens.GRCh38.81.cdna.all.fa.gz", "Homo_sapiens.RepBase.20_05.merged.fa")
erccDb<-erccDbLiteFromFasta(paste0(appSession$fastaPath, "/tmp/", "ERCC.fa.gz"))

```

```

## Extracting spike-in associations...done.
## Creating the database...done.
## Writing the spike-in tables...done.

```

```
erccPkg<-makeErccDbLitePkg(erccDb, destDir=paste0(appSession$fastaPath, "/", "tmp"))
```

```
## Creating package in /usr/local/lib/R/site-library/arkasData/extdata/fasta/tmp/ErccDbLite.ERCC.97
```

```

#Create a Ensembl Annotation Db with cdna and ncna
lapply(fastaTx, function(x) findDupes(x))

```

```
## found no duplicated sequence names .... no dupes found
```

```
## found no duplicated sequence names .... no dupes found
```

```

## Warning in .Call2("fasta_index", filexp_list, nrec, skip, seek.first.rec, :
## reading FASTA file Homo_sapiens.RepBase.20_05.merged.fa: ignored 147
## invalid one-letter sequence codes

```

```
## found no duplicated sequence names .... no dupes found
```

```

## [[1]]
##           duplicates
## ERCC.fa.gz           0
##
## [[2]]
##                               duplicates
## Homo_sapiens.GRCh38.81.cdna.all.fa.gz           0
##
## [[3]]
##                               duplicates
## Homo_sapiens.RepBase.20_05.merged.fa           0

```

```
ensDb<-ensDbLiteFromFasta("Homo_sapiens.GRCh38.81.cdna.all.fa.gz")
```

```
## Loading required package: Biostrings
```

```
## Loading required package: XVector
```

```
## Loading required package: org.Hs.eg.db
```

```
##
```

```
## Extracting transcript lengths...done.  
## Extracting transcript descriptions...done.  
## Extracting genomic coordinates...done.  
## Extracting gene and biotype associations...done.  
## Tabulating GC content...done.  
## Tabulating transcript biotypes...done.  
## Tabulating genes.....done.  
## Creating the database...done.  
## Writing the gene table...done.  
## Tabulating gene biotypes...done.  
## Writing the gene_biotype table...done.  
## Writing the tx table...done.  
## Tabulating transcript biotypes...done.  
## Writing the tx_biotype table...done.  
## Writing the biotype_class table...done.
```

```
ensPkg<-makeEnsDbLitePkg(ensDb,destDir=paste0(appSession$fastaPath,"/", "tmp"))
```

```
## Creating package in /usr/local/lib/R/site-library/arkasData/extdata fasta/tmp/EnsDbLite.Hsapiens.81
```

```
repDb<-repDbLiteFromFasta("Homo_sapiens.RepBase.20_05.merged.fa")
```

```
## Extracting repeat lengths...done.  
## Extracting repeat descriptions...done.  
## Creating the database...
```

```
## Warning in .Call2("fasta_index", filexp_list, nrec, skip, seek.first.rec, :  
## reading FASTA file Homo_sapiens.RepBase.20_05.merged.fa: ignored 147  
## invalid one-letter sequence codes
```

```
## done.
```

```
repPkg<-makeRepDbLitePkg(repDb,destDir=paste0(appSession$fastaPath,"/", "tmp"))
```

```
## Creating package in /usr/local/lib/R/site-library/arkasData/extdata fasta/tmp/RepDbLite.Hsapiens.200
```

```
ErccDbLite(erccDb)
```

```
## ErccDbLite :
## |package_name: ErccDbLite.ERCC.97
## |db_type: ErccDbLite
## |type_of_gene_id: N/A
## |created_by: TxDbLite 1.9.12
## |creation_time: Wed Apr 20 12:51:27 2016
## |organism: N/A
## |genome_build: N/A
## |source_file: /usr/local/lib/R/site-library/arkasData/extdata/fasta/tmp/ERCC.fa.gz
## | 97 spike-in controls from 4 subgroups (no known genes).
```

`EnsDbLite(ensDb)`

```
## EnsDbLite :
## |package_name: EnsDbLite.Hsapiens.81
## |db_type: EnsDbLite
## |type_of_gene_id: Ensembl Gene ID
## |created_by: TxDbLite 1.9.12
## |creation_time: Wed Apr 20 12:54:10 2016
## |organism: Homo sapiens
## |genome_build: GRCh38
## |source_file: Homo_sapiens.GRCh38.81.cdna.all.fa.gz
## | 175372 transcripts from 38530 bundles (genes).
```

`RepDbLite(repDb)`

```
## RepDbLite :
## |package_name: RepDbLite.Hsapiens.2005
## |db_type: RepDbLite
## |type_of_gene_id: RepBase identifiers
## |created_by: TxDbLite 1.9.12
## |creation_time: Wed Apr 20 12:54:11 2016
## |organism: Homo sapiens
## |genome_build: RepBase20_05
## |source_file: Homo_sapiens.RepBase.20_05.merged.fa
## | 1116 repeat exemplars from 68 repeat families (no known genes).
```

`transcripts(ErccDbLite(erccDb))`

```
## GRanges object with 97 ranges and 9 metadata columns:
##           seqnames      ranges strand | tx_length gc_content
##           <Rle> <IRanges> <Rle>   | <integer> <numeric>
## ERCC-00002 ERCC-00002 [1, 1061]   * |      1061  0.5136664
## ERCC-00003 ERCC-00003 [1, 1023]   * |      1023  0.3264907
## ERCC-00004 ERCC-00004 [1,  523]   * |       523  0.3441683
## ERCC-00007 ERCC-00007 [1, 1135]   * |      1135  0.4537445
## ERCC-00009 ERCC-00009 [1,  984]   * |       984  0.4725610
##           ...      ...      ...   ...
## ERCC-00165 ERCC-00165 [1,  872]   * |       872  0.5000000
## ERCC-00168 ERCC-00168 [1, 1024]   * |      1024  0.3417969
## ERCC-00170 ERCC-00170 [1, 1023]   * |      1023  0.3372434
## ERCC-00171 ERCC-00171 [1,  505]   * |       505  0.4772277
```

```
## ERCC_vector ERCC_vector [1, 2732] * | 2732 0.4989019
##          tx_id  gene_id gene_name  entrezid
##          <character> <integer> <integer> <integer>
## ERCC-00002 ERCC-00002      <NA>      <NA>      <NA>
## ERCC-00003 ERCC-00003      <NA>      <NA>      <NA>
## ERCC-00004 ERCC-00004      <NA>      <NA>      <NA>
## ERCC-00007 ERCC-00007      <NA>      <NA>      <NA>
## ERCC-00009 ERCC-00009      <NA>      <NA>      <NA>
## ...
## ERCC-00165 ERCC-00165      <NA>      <NA>      <NA>
## ERCC-00168 ERCC-00168      <NA>      <NA>      <NA>
## ERCC-00170 ERCC-00170      <NA>      <NA>      <NA>
## ERCC-00171 ERCC-00171      <NA>      <NA>      <NA>
## ERCC_vector ERCC_vector      <NA>      <NA>      <NA>
##          tx_biotype gene_biotype biotype_class
##          <character> <character> <character>
## ERCC-00002      SpikeIn_D      SpikeIn      SpikeIn
## ERCC-00003      SpikeIn_D      SpikeIn      SpikeIn
## ERCC-00004      SpikeIn_A      SpikeIn      SpikeIn
## ERCC-00007 SpikeIn_unannotated      SpikeIn      SpikeIn
## ERCC-00009      SpikeIn_B      SpikeIn      SpikeIn
## ...
## ERCC-00165      SpikeIn_D      SpikeIn      SpikeIn
## ERCC-00168      SpikeIn_D      SpikeIn      SpikeIn
## ERCC-00170      SpikeIn_A      SpikeIn      SpikeIn
## ERCC-00171      SpikeIn_B      SpikeIn      SpikeIn
## ERCC_vector SpikeIn_unannotated      SpikeIn      SpikeIn
## -----
## seqinfo: 97 sequences from N/A genome; no seqlengths
```

```
transcripts(EnsDbLite(ensDb))
```

```
## GRanges object with 175372 ranges and 9 metadata columns:
##          seqnames          ranges strand |
##          <Rle>          <IRanges> <Rle> |
## ENST000000000233      7 [127588345, 127591705] + |
## ENST000000000412     12 [ 8940365, 8949955] - |
## ENST000000000442     11 [ 64305578, 64316738] + |
## ENST000000001008     12 [ 2794953, 2805423] + |
## ENST000000001146      2 [ 72129238, 72148038] - |
## ...
## ENST00000634217 CHR_HSCHR11_1_CTG7 [ 2963590, 2991183] - |
## ENST00000634219 CHR_HSCHR15_4_CTG8 [ 28491559, 28494348] + |
## ENST00000634220 CHR_HSCHR7_1_CTG4_4 [103099776, 103115340] - |
## ENST00000634221 CHR_HSCHR8_9_CTG1 [ 39107956, 39151406] + |
## ENST00000634222      6 [ 36754233, 36757400] - |
##          tx_length gc_content          tx_id          gene_id
##          <integer> <numeric>          <character>          <character>
## ENST000000000233     1103 0.6092475 ENST000000000233 ENSG000000004059
## ENST000000000412     2756 0.4746009 ENST000000000412 ENSG000000003056
## ENST000000000442     2215 0.6406321 ENST000000000442 ENSG00000173153
## ENST000000001008     3732 0.5107181 ENST000000001008 ENSG000000004478
## ENST000000001146     4732 0.5680473 ENST000000001146 ENSG000000003137
## ...
```

```
## ENST00000634217      594  0.4528620 ENST00000634217 ENSG00000273562
## ENST00000634219      508  0.5688976 ENST00000634219 ENSG00000278310
## ENST00000634220     4060  0.3500000 ENST00000634220 ENSG00000275723
## ENST00000634221      571  0.4238179 ENST00000634221 ENSG00000275594
## ENST00000634222     2182  0.5077910 ENST00000634222 ENSG00000124772
##
##          gene_name      entrezid      tx_biotype
##          <character> <character>      <character>
## ENST00000000233      ARF5          381      protein_coding
## ENST00000000412      M6PR          4074     protein_coding
## ENST00000000442      ESRRA          2101     protein_coding
## ENST00000001008      FKBP4          2288     protein_coding
## ENST00000001146     CYP26B1        56603     protein_coding
##
##          ...          ...          ...          ...
## ENST00000634217      NAP1L4          4676     protein_coding
## ENST00000634219      <NA>          <NA> processed_transcript
## ENST00000634220     NAPEPLD        22236     retained_intron
## ENST00000634221      ADAM32        203102     protein_coding
## ENST00000634222      CPNE5          57699     retained_intron
##
##          gene_biotype  biotype_class
##          <character>  <character>
## ENST00000000233      protein_coding protein_coding
## ENST00000000412      protein_coding protein_coding
## ENST00000000442      protein_coding protein_coding
## ENST00000001008      protein_coding protein_coding
## ENST00000001146      protein_coding protein_coding
##
##          ...          ...          ...
## ENST00000634217      protein_coding protein_coding
## ENST00000634219 transcribed_unprocessed_pseudogene  pseudogene
## ENST00000634220      protein_coding protein_coding
## ENST00000634221      protein_coding protein_coding
## ENST00000634222      protein_coding protein_coding
## -----
## seqinfo: 288 sequences from GRCh38 genome; no seqlengths
```

```
transcripts(RepDbLite(repDb))
```

```
## GRanges object with 1116 ranges and 9 metadata columns:
##          seqnames      ranges strand | tx_length gc_content
##          <Rle> <IRanges> <Rle>   | <integer> <numeric>
##          HERVH      HERVH [1, 7713] * |      7713  0.4602619
##          X21_LINE    X21_LINE [1, 185] * |      185  0.3351351
##          UCON50      UCON50 [1, 133] * |      133  0.2105263
##          Charlie22a Charlie22a [1, 491] * |      491  0.3808554
##          PrimLTR79    PrimLTR79 [1, 503] * |      503  0.4174950
##
##          ...          ...          ...          ...          ...
##          SVA_E        SVA_E [1, 1382] * |      1382  0.6099855
##          SVA_F        SVA_F [1, 1375] * |      1375  0.6094545
##          AluYb11      AluYb11 [1, 289] * |      289  0.6332180
##          AluYb10      AluYb10 [1, 288] * |      288  0.6354167
##          AluYb8a1     AluYb8a1 [1, 287] * |      287  0.6376307
##
##          tx_id      gene_id gene_name  entrezid tx_biotype
##          <character> <integer> <integer> <integer> <character>
##          HERVH      HERVH      <NA>      <NA>      <NA>      ERV1
##          X21_LINE    X21_LINE    <NA>      <NA>      <NA>      CR1
```



```

##      UCON50      UCON50      <NA>      <NA>      <NA>      hAT
##      Charlie22a  Charlie22a  <NA>      <NA>      <NA>      hAT
##      PrimLTR79   PrimLTR79   <NA>      <NA>      <NA>      ERV1
##      ...        ...        ...        ...        ...        ...
##      SVA_E      SVA_E      <NA>      <NA>      <NA>      SVA
##      SVA_F      SVA_F      <NA>      <NA>      <NA>      SVA
##      AluYb11    AluYb11    <NA>      <NA>      <NA>      Alu
##      AluYb10    AluYb10    <NA>      <NA>      <NA>      Alu
##      AluYb8a1   AluYb8a1   <NA>      <NA>      <NA>      Alu
##      gene_biotype biotype_class
##      <character>  <character>
##      HERVH      LTR_element  repeat
##      X21_LINE    LINE        repeat
##      UCON50      DNA_element  repeat
##      Charlie22a  DNA_element  repeat
##      PrimLTR79   LTR_element  repeat
##      ...        ...        ...
##      SVA_E      other_repeat  repeat
##      SVA_F      other_repeat  repeat
##      AluYb11    SINE         repeat
##      AluYb10    SINE         repeat
##      AluYb8a1   SINE         repeat
##      -----
##      seqinfo: 1116 sequences from RepBase20_05 genome; no seqlengths

```

```

files<-dir(paste0(appSession$fastaPath,"/tmp"))[!dir(paste0(appSession$fastaPath,"/tmp")) %in% fastaTx]
lapply(files,function(x) system(paste0("rm -r ",x)))

```

```

## [[1]]
## [1] 0
##
## [[2]]
## [1] 0
##
## [[3]]
## [1] 0
##
## [[4]]
## [1] 0
##
## [[5]]
## [1] 0
##
## [[6]]
## [1] 0
##
## [[7]]
## [1] 0
##
## [[8]]
## [1] 0

```

5 Annotating Merged KallistoExperiment Containers

Arkas has a function “annotateFeatures.R” which annotates ERCC, Ensembl, and RepBase databases for species Homo-Sapiens, Mus-musculus, and Rattus norvegicus. The method “annotateFeatures.R” annotates the merged KallistoExperiment against every TxDbLite library simultaneously. These annotation databases are defined as ‘lite’ because they do not store exonic or intronic coordinates.

```
suppressPackageStartupMessages(library(arkas))
library(arkasData)
suppressPackageStartupMessages(library(TxDbLite))
samples<-c("n1","n2","n4","s1","s2","s4")
pathBase<-system.file("extdata",package="arkasData")
merged <- mergeKallisto(samples, outputPath=pathBase)
```

```
## Setting transcriptome automatically from Kallisto call string.
```

```
libraryPath<-system.file("extdata","Libraries",package="arkasData")
command<-paste0("sudo R CMD INSTALL ",libraryPath,"/",dir(libraryPath))
setwd(libraryPath)
lapply(command,function(x) system(x))
```

```
## [[1]]
## [1] 0
##
## [[2]]
## [1] 0
##
## [[3]]
## [1] 0
```

```
merged<-annotateFeatures(merged, level="transcript") #annotate features
```

```
## Loading required package: ErccDbLite.ERCC.97
```

```
## Loading required package: EnsDbLite.Hsapiens.81
```

```
## Warning in .Seqinfo.mergexy(x, y): The 2 combined objects have no sequence levels in common. (Use
##   suppressWarnings() to suppress this warning.)
```

```
## Loading required package: RepDbLite.Hsapiens.2007
```

```
## Warning in .Seqinfo.mergexy(x, y): The 2 combined objects have no sequence levels in common. (Use
##   suppressWarnings() to suppress this warning.)
```

```
NS<-suppressWarnings(annotateFeatures(NS,level="transcript"))
NS$subject <- factor(substr(colnames(NS), 2, 2))
NS$treatment <- substr(colnames(NS), 1, 1) == "s"
NS$ID <- NULL
design <- with(as(colData(NS), "data.frame"),
              model.matrix( ~ treatment + subject ))
```

```

rownames(design) <- colnames(NS)
metadata(NS)$design <- design
#returns a KallistoExperiment at the gene level
GWA<-geneWiseAnalysis(NS,design=design,
                      how="cpm",
                      p.cutoff=0.05,
                      fold.cutoff=1,
                      read.cutoff=1,
                      species="Homo.sapiens")

```

```
## Fitting bundles...
```

```
## For the time being, only summing of bundles is supported
```

```
## finding entrez IDs of top ensembl genes...
```

```
head(GWA$limmaWithMeta,n=20)
```

##		logFC	AveExpr	t	P.Value	adj.P.Val
##	ENSG00000000938	2.503936	3.4115048	3.452477	1.909930e-03	3.964324e-02
##	ENSG00000000971	-3.222736	1.4423090	-3.786282	8.123283e-04	2.105124e-02
##	ENSG00000001630	-1.822014	6.4316184	-5.308421	1.483109e-05	9.036595e-04
##	ENSG00000002822	3.750556	1.1385317	4.073160	3.850783e-04	1.189385e-02
##	ENSG00000003137	-6.183849	-1.8495805	-6.703350	4.074361e-07	5.101421e-05
##	ENSG00000003402	1.631505	8.2001798	7.894025	2.238905e-08	5.078680e-06
##	ENSG00000004478	-2.185685	3.8978332	-3.916894	5.789056e-04	1.637483e-02
##	ENSG00000005187	-1.943089	4.9011626	-4.206421	2.715437e-04	9.147383e-03
##	ENSG00000005381	2.471777	6.5678349	7.247126	1.059470e-07	1.745777e-05
##	ENSG00000005810	-1.625765	7.8181754	-6.920968	2.366541e-07	3.280570e-05
##	ENSG00000005844	2.713230	4.3581900	4.968384	3.634861e-05	1.889533e-03
##	ENSG00000006062	2.943912	4.2249563	5.153493	2.230152e-05	1.254523e-03
##	ENSG00000006118	3.585798	-0.3106464	3.415789	2.095688e-03	4.219900e-02
##	ENSG00000006125	-1.112240	6.9416204	-3.925823	5.656129e-04	1.609003e-02
##	ENSG00000006831	-2.612343	5.8762788	-6.680104	4.319203e-07	5.350451e-05
##	ENSG00000007202	1.075648	6.5488238	3.382974	2.276585e-03	4.495644e-02
##	ENSG00000007237	3.324875	3.7464312	5.593566	7.022544e-06	5.153751e-04
##	ENSG00000007384	-4.976736	-1.0944191	-4.491820	1.280544e-04	5.177461e-03
##	ENSG00000008130	2.300708	4.0822873	4.425604	1.525006e-04	5.880027e-03
##	ENSG00000008283	-2.560269	3.9195730	-4.418107	1.555460e-04	5.944953e-03
##		B	entrez_id	gene_name	ensembl_id	
##	ENSG00000000938	-1.4095287	2268	FGR	ENSG00000000938	
##	ENSG00000000971	-0.5100927	3075	CFH	ENSG00000000971	
##	ENSG00000001630	2.8192731	1595	CYP51A1	ENSG00000001630	
##	ENSG00000002822	0.1513972	8379	MAD1L1	ENSG00000002822	
##	ENSG00000003137	6.1313765	56603	CYP26B1	ENSG00000003137	
##	ENSG00000003402	9.0162886	8837	CFLAR	ENSG00000003402	
##	ENSG00000004478	-0.3685884	2288	FKBP4	ENSG00000004478	
##	ENSG00000005187	0.2188737	6296	ACSM3	ENSG00000005187	
##	ENSG00000005381	7.7419880	4353	MPO	ENSG00000005381	
##	ENSG00000005810	6.6960183	23077	MYCBP2	ENSG00000005810	
##	ENSG00000005844	2.2455964	3683	ITGAL	ENSG00000005844	
##	ENSG00000006062	2.7250540	9020	MAP3K14	ENSG00000006062	

```

## ENSG00000006118 -1.3272816 54972 TMEM132A ENSG00000006118
## ENSG00000006125 -0.8679923 163 AP2B1 ENSG00000006125
## ENSG00000006831 6.4144336 79602 ADIPOR2 ENSG00000006831
## ENSG00000007202 -2.1223845 9703 KIAA0100 ENSG00000007202
## ENSG00000007237 3.8426315 8522 GAS7 ENSG00000007237
## ENSG00000007384 1.0616307 64285 RHBDF1 ENSG00000007384
## ENSG00000008130 0.8776605 65220 NADK ENSG00000008130
## ENSG00000008283 0.8797450 1534 CYB561 ENSG00000008283
## gene_biotype biotype_class
## ENSG00000000938 protein_coding protein_coding
## ENSG00000000971 protein_coding protein_coding
## ENSG00000001630 protein_coding protein_coding
## ENSG00000002822 protein_coding protein_coding
## ENSG00000003137 protein_coding protein_coding
## ENSG00000003402 protein_coding protein_coding
## ENSG00000004478 protein_coding protein_coding
## ENSG00000005187 protein_coding protein_coding
## ENSG00000005381 protein_coding protein_coding
## ENSG00000005810 protein_coding protein_coding
## ENSG00000005844 protein_coding protein_coding
## ENSG00000006062 protein_coding protein_coding
## ENSG00000006118 protein_coding protein_coding
## ENSG00000006125 protein_coding protein_coding
## ENSG00000006831 protein_coding protein_coding
## ENSG00000007202 protein_coding protein_coding
## ENSG00000007237 protein_coding protein_coding
## ENSG00000007384 protein_coding protein_coding
## ENSG00000008130 protein_coding protein_coding
## ENSG00000008283 protein_coding protein_coding

```

6 Gene Wise Analysis

Gene wise analysis collapses transcripts into groups related to specific ensembl “gene” Ids. The package TxDbLite parses the Ensembl, or RepBase transcript fasta files and stores the respective gene id’s associated with the given transcript documented in the transcript fasta header. Arkas’ method for gene wise analysis calls “collapseBundles.R” which then calculates the aggregated total counts of transcripts for each unique gene id association. Thus the “gene” count is defined as the sum of all quantified transcripts associated with a specific gene identifier.

6.1 Understanding Gene Wise Analysis Output

The output contains a list of limma derived expression values, and enrichment data derived by biomaRt. ##Expression Results

The expression results were generated by limma/voom and have the meta biotype, gene name, etc information included in the gene wise analysis results.

6.2 Understanding Gene Wise Analysis Output

The output contains a list of limma derived expression values, and entrezID, gene name, and gene biotypes derived by biomaRt and TxDbLite respectively. The expression results were generated by limma/voom and have the meta biotype, gene name, etc information included in the gene wise analysis results.