

Genomic Features and Sequences in *Bioconductor*

Hervé Pagès

Fred Hutchinson Cancer Research Center

July 29, 2010

Combining the tools

USE CASE I: Confirmation of the GT-AG rule for Yeast

USE CASE II: Extract the Yeast transcriptome and translate it

USE CASE III: Remap probeset ids to their corresponding genes using sequence matching

Outline

Combining the tools

USE CASE I: Confirmation of the GT-AG rule for Yeast

USE CASE II: Extract the Yeast transcriptome and translate it

USE CASE III: Remap probeset ids to their corresponding genes using sequence matching

Available data

Relational annotations

- ▶ User-made transcript centric annotations: `makeTranscriptDbFromUCSC` and `makeTranscriptDbFromBiomart` functions in *GenomicFeatures*.
- ▶ Platform-specific annotation packages. E.g. *yeast2.db*.

Sequences

- ▶ Platform-specific probe packages. E.g. *yeast2probe*.
- ▶ Full genome sequences aka *BSgenome data packages*.

Available data: Platform-specific annotation packages

Mappings

```
> library(yeast2.db)
> ls('package:yeast2.db')

[1] "yeast2"                "yeast2ALIAS"
[3] "yeast2ALIAS2PROBE"     "yeast2CHR"
[5] "yeast2CHRLNGTHS"       "yeast2CHRLOC"
[7] "yeast2CHRLOCEND"       "yeast2DESCRIPTION"
[9] "yeast2ENSEMBL"         "yeast2ENSEMBL2PROBE"
[11] "yeast2ENZYME"          "yeast2ENZYME2PROBE"
[13] "yeast2GENENAME"        "yeast2GO"
[15] "yeast2GO2ALLPROBES"    "yeast2GO2PROBE"
[17] "yeast2MAPCOUNTS"      "yeast2ORF"
[19] "yeast2ORGANISM"        "yeast2ORGPKG"
[21] "yeast2PATH"            "yeast2PATH2PROBE"
[23] "yeast2PMID"            "yeast2PMID2PROBE"
[25] "yeast2_dbInfo"         "yeast2_dbconn"
[27] "yeast2_dbfile"         "yeast2_dbschema"
```

Available data: Platform-specific annotation packages

Left/Right keys

```
> Lkeys(yeast2ENSEMBL) [1:5]
```

```
[1] "1769308_at" "1769309_at" "1769310_at" "1769311_at"  
[5] "1769312_at"
```

```
> Rkeys(yeast2ENSEMBL) [1:5]
```

```
[1] "Q0045" "Q0050" "Q0055" "Q0060" "Q0065"
```

Available data: Platform-specific annotation packages

Mapped/unmapped keys

```
> mget(c("1769308_at", "1769309_at"), yeast2ENSEMBL)
```

```
$`1769308_at`
```

```
[1] "YKR009C"
```

```
$`1769309_at`
```

```
[1] NA
```

```
> mappedLkeys(yeast2ENSEMBL)[1:5]
```

```
[1] "1769308_at" "1769311_at" "1769312_at" "1769313_at"
```

```
[5] "1769314_at"
```

Available data: Platform-specific probe packages

```
> library(yeast2probe)
> yeast2probe

Object of class probetable data.frame with 120855 rows and 6 columns.

> dim(yeast2probe)
[1] 120855      6

> colnames(yeast2probe)
[1] "sequence"
[2] "x"
[3] "y"
[4] "Probe.Set.Name"
[5] "Probe.Interrogation.Position"
[6] "Target.Strandedness"

> yeast2probe$sequence[1:5]
[1] "GAAAGTTTCAGTGCACGTCTTCAAA" "GTATATTTCTAATCTTCCTCTTCAT"
[3] "ATATCAAACCGCGTACTTCGTGACT" "TAACTTTGTCTTGGATCCTGCTTTA"
[5] "ATCCGTTTTGCTGATTCCACTGATC"

> yeast2probe$Probe.Set.Name[1:5]
[1] "1769438_at" "1769438_at" "1769438_at" "1769438_at"
[5] "1769438_at"
```


Available data: BSgenome data packages

- ▶ One genome per package.
- ▶ Full genome sequences stored in Biostrings containers.
- ▶ 14 organisms / 22 packages in the current release (BioC 2.6).
- ▶ Most (but not all) packages contain sequences with builtin masks.
- ▶ Naming convention: `BSgenome.Organism.Provider.BuildVersion`

Available data: `available.genomes()`

Use the `available.genomes` function (from the *BSgenome* software package) to get the list:

```
> library(BSgenome)
> available.genomes()

[1] "BSgenome.Amellifera.BeeBase.assembly4"
[2] "BSgenome.Amellifera.UCSC.apiMel2"
[3] "BSgenome.Athaliana.TAIR.01222004"
[4] "BSgenome.Athaliana.TAIR.04232008"
[5] "BSgenome.Btaurus.UCSC.bosTau3"
[6] "BSgenome.Btaurus.UCSC.bosTau4"
[7] "BSgenome.Celegans.UCSC.ce2"
[8] "BSgenome.Cfamiliaris.UCSC.canFam2"
[9] "BSgenome.Dmelanogaster.UCSC.dm2"
[10] "BSgenome.Dmelanogaster.UCSC.dm3"
[11] "BSgenome.Drerio.UCSC.danRer5"
[12] "BSgenome.Ecoli.NCBI.20080805"
[13] "BSgenome.Ggallus.UCSC.galGal3"
[14] "BSgenome.Hsapiens.UCSC.hg17"
[15] "BSgenome.Hsapiens.UCSC.hg18"
[16] "BSgenome.Hsapiens.UCSC.hg19"
[17] "BSgenome.Mmusculus.UCSC.mm8"
[18] "BSgenome.Mmusculus.UCSC.mm9"
[19] "BSgenome.Ptroglodytes.UCSC.panTro2"
[20] "BSgenome.Rnorvegicus.UCSC.rn4"
[21] "BSgenome.Scerevisiae.UCSC.sacCer1"
[22] "BSgenome.Scerevisiae.UCSC.sacCer2"
```

Available data: Making your own BSgenome data package

It's easy to make your own package if your organism is not supported. This is documented in the BSgenomeForge vignette in the *BSgenome* software package.

Available software

Range manipulation

- ▶ *IRanges*
- ▶ *GenomicRanges*

Creation and manipulation of transcript centric annotations

- ▶ *GenomicFeatures*

Sequence manipulation

- ▶ *Biostrings*
- ▶ *BSgenome*

Outline

Combining the tools

USE CASE I: Confirmation of the GT-AG rule for Yeast

USE CASE II: Extract the Yeast transcriptome and translate it

USE CASE III: Remap probeset ids to their corresponding genes using sequence matching

The GT-AG rule

"Nearly all eukaryotic nuclear introns begin with the nucleotide sequence GT, and end with AG."

See <http://en.wikipedia.org/wiki/Intron> for more info.

Our use case is to confirm the GT-AG rule for Yeast. We choose to use the sacCer2 reference genome from UCSC for this.

What we will use

Reference genome (sequences)

The *BSgenome.Scerevisiae.UCSC.sacCer2* package.

From *GenomicFeatures*

- ▶ The `makeTranscriptDbFromUCSC` function to make the *TranscriptDb* object from the `sacCer2` genome.
- ▶ The `intronsByTranscript` function to extract the intron ranges from this *TranscriptDb* object.

From *BSgenome*

- ▶ The `getSeq` function to extract the intron sequences from *BSgenome.Scerevisiae.UCSC.sacCer2*.

What we will use (continued)

From *Biostrings*

- ▶ The `consensusMatrix` and `consensusString` functions.
- ▶ The `narrow` method for *DNASTringSet* objects to cut the region of interest from our intron sequences.

Plus...

- ▶ all the things we are going to use implicitly like the *GRanges*, *GRangesList*, *DNASTringSet* and *BSgenome* containers, and more...

Make the TranscriptDb object for sacCer2

```
> library(GenomicFeatures)
> head(supportedUCSCTables())
```

	track	subtrack
knownGene	UCSC Genes	<NA>
knownGeneOld3	Old UCSC Genes	<NA>
wgEncodeGencodeManualRel2	Gencode Genes	Genecode Manual
wgEncodeGencodeAutoRel2	Gencode Genes	Genecode Auto
wgEncodeGencodePolyaRel2	Gencode Genes	Genecode PolyA
ccdsGene	Consensus CDS	<NA>

Make the TranscriptDb object for sacCer2 (continued)

```
> rownames(supportedUCSCTables())
```

[1]	"knownGene"	"knownGeneOld3"
[3]	"wgEncodeGencodeManualRel2"	"wgEncodeGencodeAutoRel2"
[5]	"wgEncodeGencodePolyaRel2"	"ccdsGene"
[7]	"refGene"	"xenoRefGene"
[9]	"vegaGene"	"vegaPseudoGene"
[11]	"ensGene"	"acembly"
[13]	"sibGene"	"nscanPasaGene"
[15]	"nscanGene"	"sgdGene"
[17]	"sgpGene"	"geneid"
[19]	"genscan"	"exoniphy"
[21]	"augustusHints"	"augustusXRA"
[23]	"augustusAbinitio"	"acescan"

Make the TranscriptDb object for sacCer2 (continued)

```
> txdb1 <- makeTranscriptDbFromUCSC(genome="sacCer2", tablename="sgdGene")
```

Make the TranscriptDb object for sacCer2 (continued)

```
> txdb1
```

```
TranscriptDb object:
```

```
| Db type: TranscriptDb  
| Data source: UCSC  
| Genome: sacCer2  
| UCSC Table: sgdGene  
| Type of Gene ID: ID of canonical transcript in cluster  
| Full dataset: yes  
| transcript_nrow: 6717  
| exon_nrow: 7083  
| cds_nrow: 7061  
| Db created by: GenomicFeatures package from Bioconductor  
| Creation time: 2010-07-29 11:06:56 -0700 (Thu, 29 Jul 2010)  
| GenomicFeatures version at creation time: 1.0.6  
| RSQLite version at creation time: 0.9-2
```

Extract the introns ranges

```
> introns <- intronsByTranscript(txdb1)
> introns

GRangesList of length 6717
$1
GRanges with 0 ranges and 0 elementMetadata values
  seqnames ranges strand |

$2
GRanges with 0 ranges and 0 elementMetadata values
  seqnames ranges strand |

$3
GRanges with 0 ranges and 0 elementMetadata values
  seqnames ranges strand |

...
<6714 more elements>

seqlengths
  chrIV  chrXV  chrVII  chrXII ...   chrI    chrM 2micron
1531919 1091289 1090947 1078175 ... 230208 85779   6318
```

A quick look at this GRangesList object

Nb of introns per transcript

```
> table(elementLengths(introns))
```

0	1	2	3	4	5	7
6365	334	13	2	1	1	1

Total nb of introns

```
> sum(elementLengths(introns))
```

```
[1] 382
```

From GRangesList to GRanges (unlist)

```
> introns <- unlist(introns)
> introns
```

GRanges with 382 ranges and 0 elementMetadata values

	seqnames	ranges	strand	
	<Rle>	<IRanges>	<Rle>	
53.1	chrI	[87389, 87501]	+	
82.2	chrI	[142256, 142621]	+	
84.3	chrI	[151009, 151098]	-	
125.4	chrM	[13987, 16434]	+	
125.5	chrM	[16471, 18953]	+	
125.6	chrM	[18992, 20507]	+	
125.7	chrM	[20985, 21994]	+	
125.8	chrM	[22247, 23611]	+	
125.9	chrM	[23747, 25317]	+	
...
6507.374	chrXIII	[550801, 551202]	-	
6509.375	chrXIII	[551951, 552494]	+	
6568.376	chrXIII	[651160, 651622]	+	
6569.377	chrXIII	[652775, 652846]	-	
6576.378	chrXIII	[666933, 667016]	-	
6600.379	chrXIII	[721198, 721344]	-	
6605.380	chrXIII	[732465, 732874]	+	
6618.381	chrXIII	[753742, 754218]	-	
6672.382	chrXIII	[854816, 854897]	+	

Load the sacCer2 genome

It's important to use the same reference genome as for the *TranscriptDb* object.

```
> library(BSgenome.Scerevisiae.UCSC.sacCer2)
```


A quick look at the sacCer2 genome

```
> Scerevisiae
```

```
Yeast genome
```

```
|  
| organism: Saccharomyces cerevisiae (Yeast)  
| provider: UCSC  
| provider version: sacCer2  
| release date: June 2008  
| release name: SGD June 2008 sequence  
|  
| sequences (see '?seqnames'):  
|   chrI      chrII      chrIII      chrIV      chrV      chrVI  
|   chrVII     chrVIII     chrIX      chrX      chrXI     chrXII  
|   chrXIII    chrXIV     chrXV      chrXVI     chrM      2micron  
|  
| (use the '$' or '[' operator to access a given sequence)
```

```
> Scerevisiae$chrI
```

```
230208-letter "DNAString" instance
```

```
seq: CCACACCACCCCACACCCCACACAC...GTGTGGTGTGGGTGTGGTGTGTGTGGG
```

```
> seqlengths(Scerevisiae)
```

chrI	chrII	chrIII	chrIV	chrV	chrVI	chrVII
230208	813178	316617	1531919	576869	270148	1090947
chrVIII	chrIX	chrX	chrXI	chrXII	chrXIII	chrXIV
562643	439885	745742	666454	1078175	924429	784333

A quick look at the sacCer2 genome (continued)

```
> seqlengths(Scerevisiae)
```

chrI	chrII	chrIII	chrIV	chrV	chrVI	chrVII
230208	813178	316617	1531919	576869	270148	1090947
chrVIII	chrIX	chrX	chrXI	chrXII	chrXIII	chrXIV
562643	439885	745742	666454	1078175	924429	784333
chrXV	chrXVI	chrM	2micron			
1091289	948062	85779	6318			

Extract the intron sequences

```
> intron_seqs <- getSeq(Scerevisiae, introns, as.character=FALSE)
```

A quick look at this DNASTringSet object

```
> intron_seqs
```

```
A DNASTringSet instance of length 382
```

```
      width seq
[1]    113 GTAAGTACAGAAAGCCACAGAGTA...ACGTTCTTCGTGTTTATTTTTAG
[2]    366 GTATGTTCCGATTTAGTTTACTTT...TTTTGTTTCTCCTTTTAAATAG
[3]     90 GTATGTTTCATGTCTCATTCTCCTT...TATTTACTAACGACACATTGAAG
[4]   2448 GTGCGCCTCTCAGTGCGTATATTT...CATAGGTTAATTTGCTATTTTCAT
[5]   2483 GTGCGCCGTTTCGCTTAATTTATC...TTCAGATAGGTTTGCTACTCTAC
[6]   1516 CAAAAAAGATATGAAAGTAATAAT...TGAAAGATTATAATAAAATGAAC
[7]   1010 CAAACAGTGGCCCTTATTATTATA...ATAATATATATATATATAACAAG
[8]   1365 ATAAATCCCTTTAGCAAGGATAAA...ATGTTTTAAAGTTAAATAAAAGA
[9]   1571 ATTAATTTAATAAGTGTCGTGCTT...AATAATATTCTTTTTTTTTTATG
...
...
[374]   402 GTATGTTGAACTGAAGCAATAAGA...AACTGATTTTTTTTATGATTATAG
[375]   544 GTATGTTTTTCAGTTCTGCAGAATG...AACTAATTGCATTACTTCTTTAG
[376]   463 GTATGTGAGACATAAACAAGGAAC...ACAACCTGTGTCCTTATATTTAG
[377]    72 GTTTGTAATATTAACCTCAAAAAGA...ACGTTTTTTTCACATTAATTTTAG
[378]    84 GTATGTGTGAAAATGATTCTGTGT...TAACGATGAGATGAGCTGTGCAG
[379]   147 GTATGTATTTTTTTTTTCGCTCTGTT...TATGGTCATATCATTGATTTTCAG
[380]   410 GTATGTTTGCATTTTTTTAGGTGAA...ATTACGATCGCATATCGAAATAG
[381]   477 GTGAGTAAATACCTACTAAACTAT...GAAAATCCTTGTTATTTTATCAG
[382]    82 GTATGTTTTTAATATTTTAGATGC...ACTAACAACCTTACTTTTCACTAG
```

Look at the first 2 bases of each intron

```
> narrow(intron_seqs, end=2) # error!
```

```
> table(width(intron_seqs))
```

1	2	3	5	7	10	12	13	15	18	27	31
47	1	1	1	5	1	3	2	1	1	1	1
35	39	40	49	52	54	56	58	59	62	63	65
1	2	1	1	1	1	2	1	1	2	2	1
67	68	69	70	71	72	73	74	75	76	77	78
2	2	2	3	2	2	1	1	3	3	2	1
79	80	81	82	83	84	85	86	87	88	89	90
1	5	1	3	6	3	3	5	3	4	3	4
91	92	93	94	95	96	97	99	100	101	102	103
2	3	5	3	2	4	3	7	1	2	2	1
104	105	106	108	110	111	113	114	116	118	119	122
1	2	1	1	1	3	3	1	5	1	1	1
123	124	126	128	131	133	134	139	141	143	147	148
2	1	1	1	1	1	2	1	2	2	2	4
149	152	156	162	168	179	194	200	209	213	230	238
1	2	1	1	1	1	1	1	1	1	1	1
252	256	268	269	273	275	279	290	292	298	301	306
1	2	1	1	1	1	2	1	1	1	1	1
307	308	314	317	320	321	322	326	330	339	342	345
1	1	1	1	1	1	2	1	1	1	1	1
347	349	350	352	357	359	362	365	366	368	383	384
1	1	1	1	1	2	1	2	1	1	1	3
386	388	389	390	394	397	398	400	401	402	403	405

Look at the first 2 bases of each intron (continued)

```
> narrow(intron_seqs, end=2)
A DNAStringSet instance of length 333
      width seq
[1]      2 GT
[2]      2 GT
[3]      2 GT
[4]      2 GT
[5]      2 GT
[6]      2 CA
[7]      2 CA
[8]      2 AT
[9]      2 AT
...      ...
[325]     2 GT
[326]     2 GT
[327]     2 GT
[328]     2 GT
[329]     2 GT
[330]     2 GT
[331]     2 GT
[332]     2 GT
[333]     2 GT
```

Consensus matrix

```
> consensusMatrix(narrow(intron_seqs, end=2), baseOnly=TRUE)
```

	[,1]	[,2]
A	17	21
C	10	14
G	297	3
T	9	295
other	0	0

Consensus string

```
> consensusString(narrow(intron_seqs, end=4))  
[1] "GTAT"
```

Looking at the last 2 bases of each intron is left as an exercise. Tip: `narrow` accepts negative start values (they are counted from the end of each sequence).

Outline

Combining the tools

USE CASE I: Confirmation of the GT-AG rule for Yeast

USE CASE II: Extract the Yeast transcriptome and translate it

USE CASE III: Remap probeset ids to their corresponding genes using sequence matching

Goal

In this use case, we want to extract the Yeast transcriptome based on the transcripts information stored in our *TranscriptDb* object.

Again, we must be careful to use the reference genome that matches exactly our *TranscriptDb* object.

What we will use

Reference genome (sequences)

The *BSgenome.Scerevisiae.UCSC.sacCer2* package.

From *GenomicFeatures*

- ▶ The `makeTranscriptDbFromUCSC` function to make the *TranscriptDb* object from the *sacCer2* genome.
- ▶ The `extractTranscriptsFromGenome` function to extract the transcriptome.
- ▶ The `cdsBy` function to extract the CDS ranges (grouped by transcripts) from this *TranscriptDb* object.

From *Biostrings*

- ▶ The `translate` function to translate the transcriptome.

Extract the full transcriptome

We assume we've already made the `txdb1` object and loaded the `sacCer2` genome like in USE CASE I.

```
> tx_seqs <- extractTranscriptsFromGenome(Scerevisiae, txdb1)
> tx_seqs
```

A DNAStringSet instance of length 6717

	width	seq	names
[1]	1185	ATGACTCTACAAG...GCCACCAACTAA	YAL012W
[2]	315	ATGATCGTAAATA...AATATTGTATAA	YAL069W
[3]	255	ATGCACGGCACTT...AATAATACATAA	YAL068W-A
[4]	363	ATGGTCAAATTAA...ATCGCAAACCTAG	YAL068C
[5]	228	ATGCCAATTATAG...GTATACTGTTAG	YAL067W-A
[6]	1782	ATGTATTCAATTG...GATGAAAAATAA	YAL067C
[7]	309	ATGTTATCTCTTG...TCATGTATATAG	YAL066W
[8]	387	ATGAACAGTGCTA...ATCGTATGGTAA	YAL065C
[9]	381	ATGGCAGGTGAAG...GTGCACACATGA	YAL064W-B
...	
[6709]	714	ATGACTCCAAAAA...TTATATAGCTGA	YMR322C
[6710]	1314	ATGTCCATCACGA...AACAAACTATAA	YMR323W
[6711]	243	ATGATCACTATGA...GCGTATTGCTAA	YMR324C
[6712]	375	ATGGTCAAATTAA...ATTCCAAAATAG	YMR325W
[6713]	309	ATGCACGGCACTT...ACAGTTACATAA	YMR326C
[6714]	1272	ATGCCACAATTTG...AGACGCATATAA	R0010W
[6715]	1122	ATGAATGGCGAGA...GTGGATGGGTAG	R0020C
[6716]	546	ATGCCTTATAAAA...GAACCTGTATAA	R0030W
[6717]	891	ATGGACGACATTG...TCTAGGGTATGA	R0040C

Translation - 1st attempt

```
> translate(tx_seqs)
```

```
A AAStringSet instance of length 6717
```

```
width seq
[1] 395 MTLQESDKFATKAIHAGEHVDVH...VGIEDTDDLLEDIKQALKQATN*
[2] 105 MIVNNTHTVLTLPYTTTTCHTHP...PITPIIIHILISISHSAVPNIV*
[3] 85 MHGTCLSGLYPVPFTHNAHHYPH...ITEKSPQKSPKHKNILLFNNNT*
[4] 121 MVKLTSIAAGVAAIAATASATTT...SSRLKPAISSALSKDGIYTIAN*
[5] 76 MPIIGVPRCLIKPFSVPVTFPFS...RRKYFHLLNSYNIKRVLGVVYC*
[6] 594 MYSIVKEIIVDPYKRLKWGFIPV...SKHGVEKPTSKDVETLSVSDEK*
[7] 103 MLSLVKRSILHSIPITRHILPIQ...IYIKEIQTKMLEKHTASDTSCI*
[8] 129 MNSATSETTTNTGAAETTTSTGA...NGLLTNNGISVFISTVLLAIVW*
[9] 127 MAGEAVSEHTPDSQEVTVTSVVC...VAPLTVTVAVETIAEEMDSVHT*
...
[6709] 238 MTPKRALISLTSYHGPFYKDGAK...VTGVNANSSYSTTIRAINALYS*
[6710] 438 MSITKVHARTVYDSRGNPTVEVE...IEEELGDDCIYAGHRFHDGNKL*
[6711] 81 MITMKMFLFLNEACIFIDSVCEG...SIFGVAECLNLVAIDPRSEAYC*
[6712] 125 MVKLTSIAAGVAAIAAGVAAAPA...TRLRPAISSALSKDGIYTAIPK*
[6713] 103 MHGTCLSGLYPVPFTHKAHDYPH...LYKINILTCGHYPLNSIPFTVT*
[6714] 424 MPQFGILCKTPPKVLVRQFVERF...AWNGIISQEVLDYLSSYINRRI*
[6715] 374 MNGERLLACIKQCIMQHFQPMVY...HWKPVDVEVEFRCKFKERKVDG*
[6716] 182 MPYKTAIDCIEELATQCFLSKLT...SLNFEHPNLGVFPETDSIFEPV*
[6717] 297 MDDIETAKNLTVKARTAYSVWDV...PTKKRRVATRVGRKSRNTSRV*
```

Translation - 1st attempt (continued)

```
> warnings()
```

```
NULL
```

Translation - 1st attempt (continued)

```
> narrow(translate(tx_seqs), start=-5)
A AAStringSet instance of length 6717
      width seq
[1]      5 QATN*
[2]      5 PNIV*
[3]      5 NNNT*
[4]      5 TIAN*
[5]      5 VVYC*
[6]      5 SDEK*
[7]      5 TSCI*
[8]      5 AIVW*
[9]      5 SVHT*
...      ...
[6709]     5 ALYS*
[6710]     5 GNKL*
[6711]     5 EAYC*
[6712]     5 AIPK*
[6713]     5 FTVT*
[6714]     5 NRRI*
[6715]     5 KVDG*
[6716]     5 FEPV*
[6717]     5 TSRV*
```

Translation - 1st attempt (continued)

```
> consensusMatrix(narrow(translate(tx_seqs), start=-5))
```

	[,1]	[,2]	[,3]	[,4]	[,5]
*	5	1	0	0	6701
A	364	316	280	338	0
C	110	99	98	114	1
D	348	353	347	329	1
E	456	429	367	417	1
F	323	375	390	345	0
G	333	314	309	170	0
H	169	143	177	201	0
I	341	439	441	483	3
K	694	577	720	753	0
L	562	688	622	706	4
M	124	159	123	147	0
N	386	391	365	462	2
P	303	243	195	158	1
Q	213	268	265	299	0
R	366	331	450	333	0
S	581	546	577	476	0
T	357	311	369	263	0
V	345	377	309	352	2
W	92	104	67	117	0
Y	245	253	246	254	1

Extract the translated part of the transcriptome

```
> cds <- cdsBy(txdb1)
> cds
```

GRangesList of length 6717

\$1

GRanges with 1 range and 3 elementMetadata values

	seqnames	ranges	strand	cds_id	cds_name
	<Rle>	<IRanges>	<Rle>	<integer>	<character>
[1]	chrI	[130802, 131986]	+	1	NA
	exon_rank				
	<integer>				
[1]	1				

\$2

GRanges with 1 range and 3 elementMetadata values

	seqnames	ranges	strand	cds_id	cds_name
	<Rle>	<IRanges>	<Rle>	<integer>	<character>
[1]	chrI	[335, 649]	+	2	NA
	exon_rank				
	<integer>				
[1]	1				

\$3

GRanges with 1 range and 3 elementMetadata values

	seqnames	ranges	strand	cds_id	cds_name
	<Rle>	<IRanges>	<Rle>	<integer>	<character>
[1]	chrI	[130802, 131986]	+	1	NA
	exon_rank				
	<integer>				
[1]	1				

Extract the translated part of the transcriptome (continued)

```
> cds_seqs <- extractTranscriptsFromGenome(Scerevisiae, cds)
> cds_seqs
```

A DNAStringSet instance of length 6717

	width	seq	names
[1]	1185	ATGACTCTACAAG...GCCACCAACTAA	1
[2]	315	ATGATCGTAAATA...AATATTGTATAA	2
[3]	255	ATGCACGGCACTT...AATAATACATAA	3
[4]	363	ATGGTCAAATTAA...ATCGCAAACCTAG	4
[5]	228	ATGCCAATTATAG...GTATACTGTTAG	5
[6]	1782	ATGTATTCAATTG...GATGAAAAATAA	6
[7]	309	ATGTTATCTCTTG...TCATGTATATAG	7
[8]	387	ATGAACAGTGCTA...ATCGTATGGTAA	8
[9]	381	ATGGCAGGTGAAG...GTGCACACATGA	9
...	
[6709]	714	ATGACTCCAAAAA...TTATATAGCTGA	6709
[6710]	1314	ATGTCCATCACGA...AACAACTATAA	6710
[6711]	243	ATGATCACTATGA...GCGTATTGCTAA	6711
[6712]	375	ATGGTCAAATTAA...ATTCCAAAATAG	6712
[6713]	309	ATGCACGGCACTT...ACAGTTACATAA	6713
[6714]	1272	ATGCCACAATTTG...AGACGCATATAA	6714
[6715]	1122	ATGAATGGCGAGA...GTGGATGGGTAG	6715
[6716]	546	ATGCCTTATAAAA...GAACCTGTATAA	6716
[6717]	891	ATGGACGACATTG...TCTAGGGTATGA	6717

Translation - 2nd attempt

```
> translate(cds_seqs)
```

```
A AAStringSet instance of length 6717
```

```
      width seq
[1]   395 MTLQESDKFATKAIHAGEHVDVH...VGIEDTDDLLEDIKQALKQATN*
[2]   105 MIVNNTHTVLTLPYTTTTCHTHP...PITPIIIHILISISHSAVPNIV*
[3]    85 MHGTCLSGLYPVPFTHNAHHYPH...ITEKSPQKSPKHKNILLFNNNT*
[4]   121 MVKLTSIAAGVAAIAATASATTT...SSRLKPAISSALSKDGIYTIAN*
[5]    76 MPIIGVPRCLIKPFSVPVTFPFS...RRKYFHLLNSYNIKRVLGVVYC*
[6]   594 MYSIVKEIIVDPYKRLKWGFIPV...SKHGVEKPTSKDVETLSVSDEK*
[7]   103 MLSLVKRSILHSIPITRHILPIQ...IYIKEIQTKMLEKHTASDTSCI*
[8]   129 MNSATSETTTNTGAAETTTSTGA...NGLLTNNGISVFISTVLLAIVW*
[9]   127 MAGEAVSEHTPDSQEVTVTSVVC...VAPLTVTVAVETIAEEMDSVHT*
...
[6709] 238 MTPKRALISLTSYHGPFYKDGAK...VTGVNANSSYSTTIRAINALYS*
[6710] 438 MSITKVHARTVYDSRGNPTVEVE...IEEELGDDCIYAGHRFHDGNKL*
[6711]  81 MITMKMFLFLNEACIFIDSVCEG...SIFGVAECLNLVAIDPRSEAYC*
[6712] 125 MVKLTSIAAGVAAIAAGVAAAPA...TRLRPAISSALSKDGIYTAIPK*
[6713] 103 MHGTCLSGLYPVPFTHKAHDYPH...LYKINILTCGHYPLNSIPFTVT*
[6714] 424 MPQFGILCKTPPKVLVRQFVERF...AWNGIISQEVLDYLSSYINRRI*
[6715] 374 MNGERLLACIKQCIMQHFQPMVY...HWKPVDVEVEFRCKFKERKVDG*
[6716] 182 MPYKTAIDCIEELATQCFLSKLT...SLNFEHPNLGVFPETDSIFEPV*
[6717] 297 MDDIETAKNLTVKARTAYSVWDV...PTKKRRVATRVGRKSRNTSRV*
```

Translation - 2nd attempt (continued)

```
> narrow(translate(cds_seqs), start=-5)
```

```
A AAStringSet instance of length 6717
```

```
width seq
```

```
[1]      5 QATN*
```

```
[2]      5 PNIV*
```

```
[3]      5 NNNT*
```

```
[4]      5 TIAN*
```

```
[5]      5 VVYC*
```

```
[6]      5 SDEK*
```

```
[7]      5 TSCI*
```

```
[8]      5 AIVW*
```

```
[9]      5 SVHT*
```

```
... ..
```

```
[6709]    5 ALYS*
```

```
[6710]    5 GNKL*
```

```
[6711]    5 EAYC*
```

```
[6712]    5 AIPK*
```

```
[6713]    5 FTVT*
```

```
[6714]    5 NRRI*
```

```
[6715]    5 KVDG*
```

```
[6716]    5 FEPV*
```

```
[6717]    5 TSRV*
```

Translation - 2nd attempt (continued)

```
> consensusMatrix(narrow(translate(cds_seqs), start=-5))
```

	[,1]	[,2]	[,3]	[,4]	[,5]
*	3	1	0	0	6716
A	366	317	282	338	0
C	108	98	98	114	0
D	348	353	348	329	0
E	457	428	367	416	1
F	323	376	391	345	0
G	335	314	309	170	0
H	170	143	177	201	0
I	341	439	438	482	0
K	696	581	723	756	0
L	563	689	618	707	0
M	124	159	123	148	0
N	387	391	364	462	0
P	300	243	195	156	0
Q	213	267	263	296	0
R	365	329	449	335	0
S	578	545	577	475	0
T	358	311	370	263	0
V	347	376	309	352	0
W	90	104	67	117	0
Y	245	253	249	255	0

Translation - 2nd attempt (continued)

As an extra sanity check, we use the `vcountPattern` function from the *Biostrings* package to count the number of `*` in each translated transcript.

```
> table(vcountPattern("*", translate(cds_seqs)))
```

	1	2	3	4	5	6	7	8	9	10	11	12
6692	5	1	3	2	3	1	2	2	2	1	2	
14												
1												

Things still don't look completely right :-/

Outline

Combining the tools

USE CASE I: Confirmation of the GT-AG rule for Yeast

USE CASE II: Extract the Yeast transcriptome and translate it

USE CASE III: Remap probeset ids to their corresponding genes using sequence matching

Motivation

The mapping from probeset ids to gene ids provided by the microarray manufacturer may not always be accurate. In this use case, we will compute this mapping using pattern matching facilities available in the *Biostrings* package to match the probe sequences against the transcriptome. Then we show how to infer the mapping from probeset ids to transcript ids from the result of this matching.

What we will use

Probe sequences for the Yeast Genome 2.0 Array

The *yeast2probe* package.

Reference genome (sequences)

The *BSgenome.Scerevisiae.UCSC.sacCer2* package.

From *GenomicFeatures*

- ▶ The `makeTranscriptDbFromUCSC` function to make the *TranscriptDb* object from the *sacCer2* genome.
- ▶ The `extractTranscriptsFromGenome` function to extract the transcriptome.

From *Biostrings*

- ▶ The `DNASTringSet` and `PDict` constructors.
- ▶ The `vwhichPDict` function to find which probes hit each transcript.

But first we have a quick look at the *yeast2.db* package.

```
> library(yeast2.db) # Affymetrix Yeast Genome 2.0 Array  
> ls('package:yeast2.db')
```

```
[1] "yeast2"                "yeast2ALIAS"  
[3] "yeast2ALIAS2PROBE"    "yeast2CHR"  
[5] "yeast2CHRLNGTHS"      "yeast2CHRLOC"  
[7] "yeast2CHRLOCEND"      "yeast2DESCRIPTION"  
[9] "yeast2ENSEMBL"        "yeast2ENSEMBL2PROBE"  
[11] "yeast2ENZYME"         "yeast2ENZYME2PROBE"  
[13] "yeast2GENENAME"       "yeast2G0"  
[15] "yeast2G02ALLPROBES"   "yeast2G02PROBE"  
[17] "yeast2MAPCOUNTS"     "yeast2ORF"  
[19] "yeast2ORGANISM"       "yeast2ORGPKG"  
[21] "yeast2PATH"           "yeast2PATH2PROBE"  
[23] "yeast2PMID"           "yeast2PMID2PROBE"  
[25] "yeast2_dbInfo"        "yeast2_dbconn"  
[27] "yeast2_dbfile"        "yeast2_dbschema"
```

yeast2.db (continued)

A sanity check:

```
> all(Rkeys(yeast2ENSEMBL) %in% names(tx_seqs))
```

```
[1] TRUE
```

Matching yeast2probe

```
> library(yeast2probe)
> yeast2_dict <- DNASTringSet(yeast2probe)
> yeast2_dict
```

```
A DNASTringSet instance of length 120855
width seq
```

```
[1] 25 GAAAGTTTCAGTGCACGTCTTCAAA
[2] 25 GTATATTTCTAATCTTCCTCTTCAT
[3] 25 ATATCAAACCGCGTACTTCGTGACT
[4] 25 TAACTTTGTCTTGGATCCTGCTTTA
[5] 25 ATCCGTTTTGCTGATTCCACTGATC
[6] 25 AAGATTATGGCGTGCTCGTGAATAC
[7] 25 GTTCGCAAATAACTCTATGCCCTCT
[8] 25 GCCATTGGAGTCGAACACAGTCTAT
[9] 25 AGTCTATCAACATTCACCCACTTAT
...
[120847] 25 GACAGCATCCTTGAATATGTAAAAG
[120848] 25 ACGAAGCCGACATGCTGTTCTCTGT
[120849] 25 TGCTGTTCTCTGTCACTGTTCCCGG
[120850] 25 GCTTTGATTCAAGTCGGAATGGCGCT
[120851] 25 TCGGAATGGCGCTCAGCAGATATTT
[120852] 25 CAGATATTTGAAGCTGACCGTCTTT
[120853] 25 TGACCGTCTTTGAAAGCGACAAATG
[120854] 25 CAGATAACCTGATCTACCAAGTGGC
[120855] 25 CTCCTGTCCATGTGAAGGTGTGGAG
```

Matching yeast2probe (continued)

```
> yeast2_pdict <- PDict(yeast2_dict)
> yeast2_pdict
```

TB_PDict object of length 120855 and width 25 (preprocessing algo="ACtree2")

```
> tx2probes <- vwhichPDict(yeast2_pdict, tx_seqs)
```