EMBL Advanced Course RNA-Seq and ChiP-Seq Data

Nicolas Delhomme, June 20th-22nd 2011, Heidelberg

EMBL

Outline

- Sequence alignment
- Aligners
- Recent development
- Aligners’ usage
- Alignment pitfall
- Bioconductor

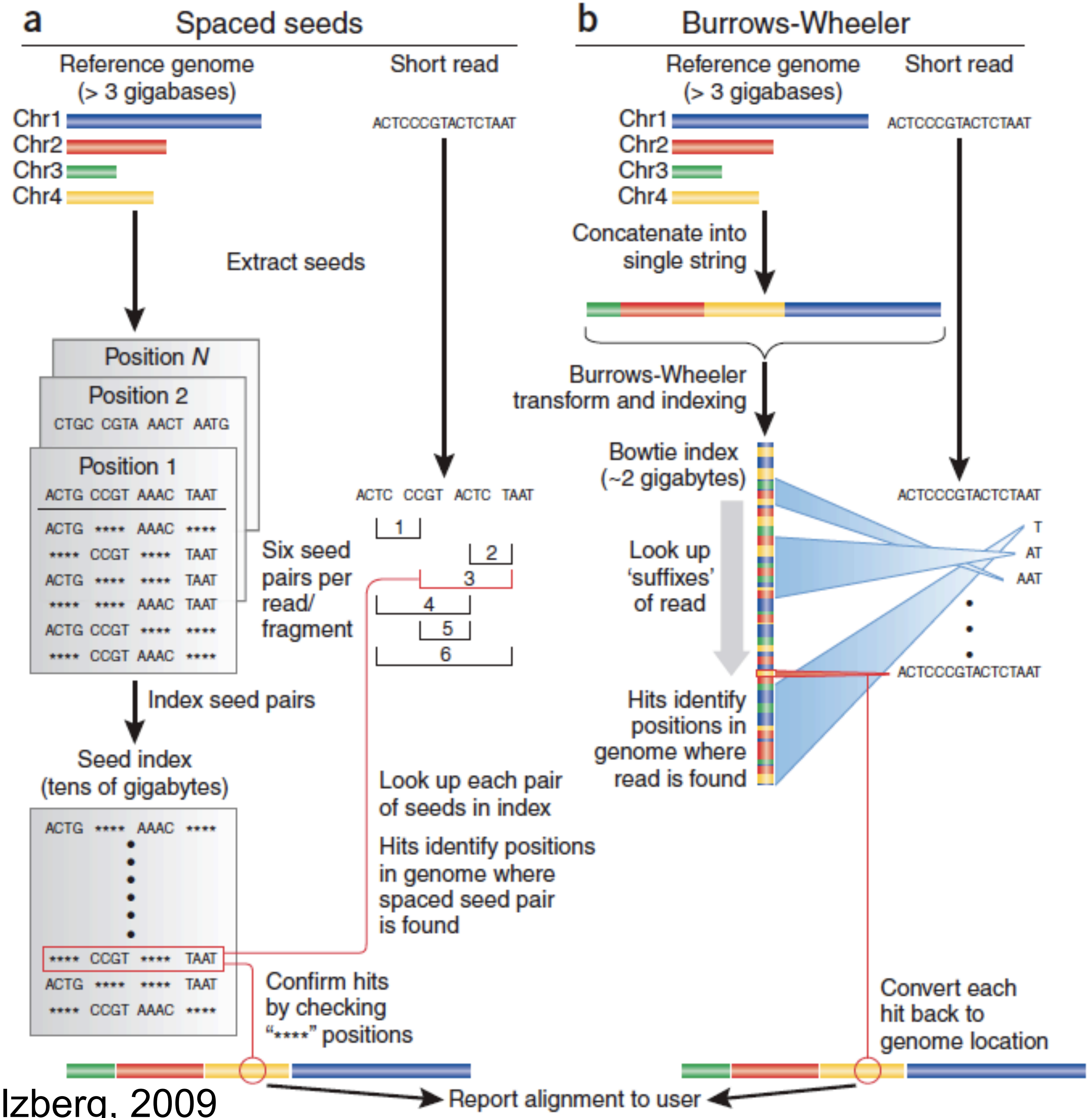
Who are we?

- Me:
  - Staff member of the Functional Genomic Center
    - Genome Biology Unit, EMBL, Heidelberg
    - co-directed by Eileen Furlong and Lars Steinmetz
  - Position 50% service, 50% research
    - service: establishment of a LIMS and pre-processing system for NGS data
    - research: analyses of NGS data of various kinds: RNAseq, TagSeq, ChIPseq (TF and Histones) and *de-novo* assembly, mainly using R
- You:
  - your aligner's knowledge?

Sequence alignment

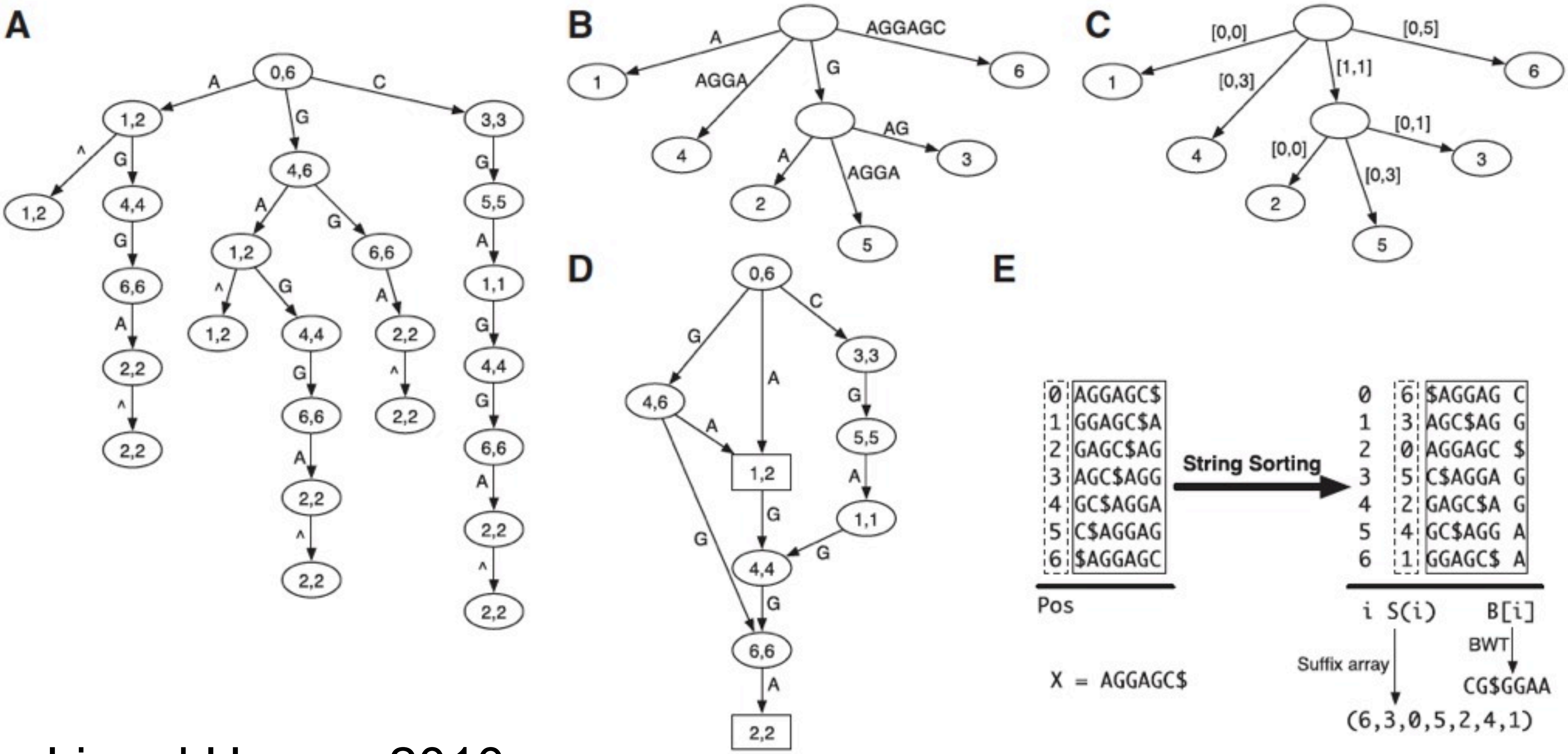
- Two main approaches:
  - based on hash table
    - spaced seeds
  - based on suffix/prefix tries
    - Burrows-Wheeler transform (BWT)

- Reviewed in Li and Homer: A survey of sequence alignment algorithms for next-generation sequencing. Briefings in Bioinformatics (2010)



Trapnell and Salzberg, 2009

Suffix/Prefix Tries



Pos	
0	AGGAGC$
1	GGAGC$A
2	GAGC$AG
3	AGC$AGG
4	GC$AGGA
5	C$AGGAG
6	$AGGAGC

X = AGGAGC$

i	S(i)		B[i]
0	6	$AGGAG	C
1	3	AGC$AG	G
2	0	AGGAGC	$
3	5	C$AGGA	G
4	2	GAGC$A	G
5	4	GC$AGG	A
6	1	GGAGC$	A

(6,3,0,5,2,4,1)

CG$GGAA

Li and Homer, 2010

Aligners

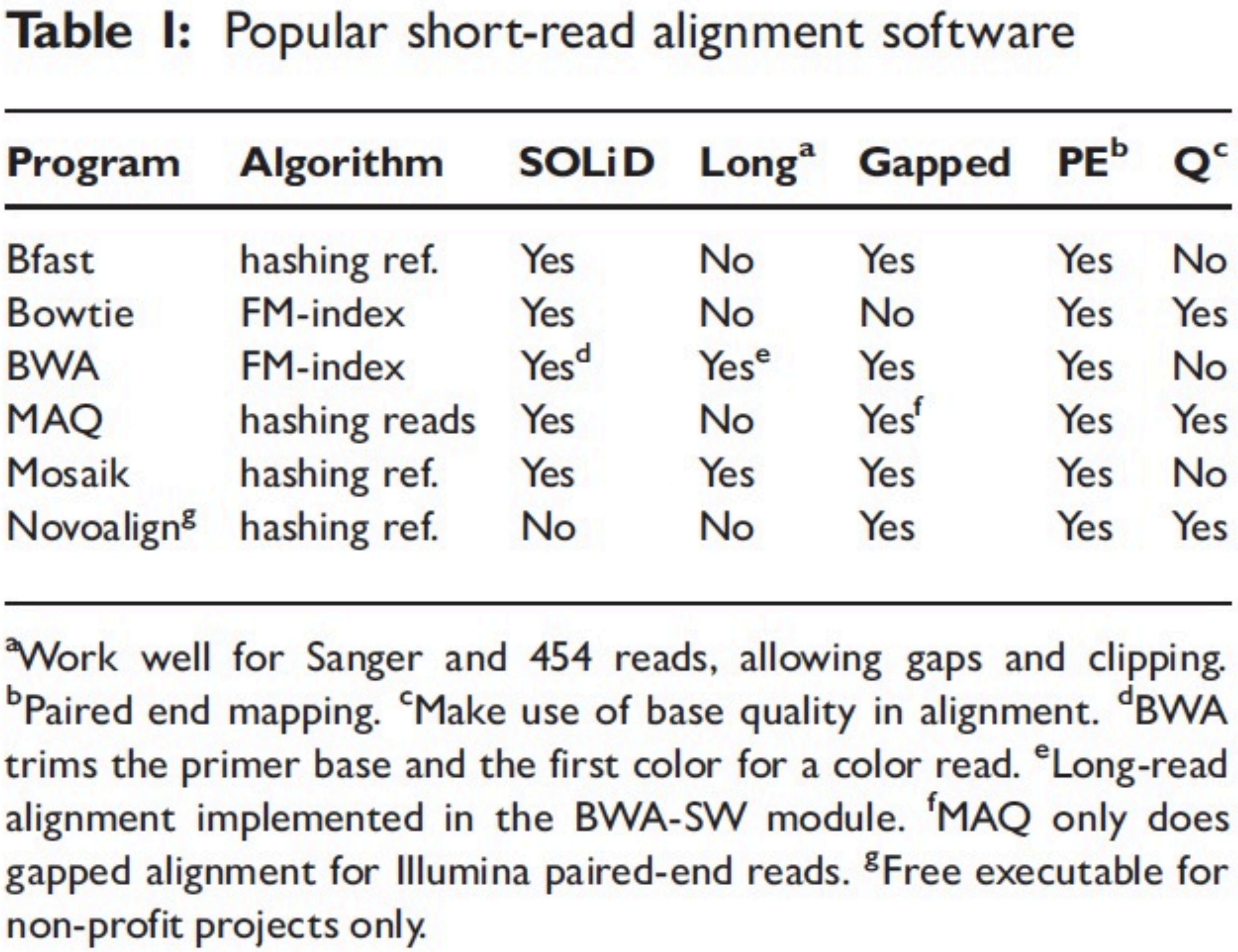


Table I: Popular short-read alignment software

Program	Algorithm	SOLiD	Long[a]	Gapped	PE[b]	Q[c]
Bfast	hashing ref.	Yes	No	Yes	Yes	No
Bowtie	FM-index	Yes	No	No	Yes	Yes
BWA	FM-index	Yes[d]	Yes[e]	Yes	Yes	No
MAQ	hashing reads	Yes	No	Yes[f]	Yes	Yes
Mosaik	hashing ref.	Yes	Yes	Yes	Yes	No
Novoalign[g]	hashing ref.	No	No	Yes	Yes	Yes

[a] Work well for Sanger and 454 reads, allowing gaps and clipping. [b] Paired end mapping. [c] Make use of base quality in alignment. [d] BWA trims the primer base and the first color for a color read. [e] Long-read alignment implemented in the BWA-SW module. [f] MAQ only does gapped alignment for Illumina paired-end reads. [g] Free executable for non-profit projects only.

Aligners c’ed

- 20 aligners published in the last 2 years
- Most deal with short reads
- some of those with ABI specific “color-space”
- A large scale study comparing them is underway:
    - GSNAP: http://research-pub.gene.com/gmap/ is the most efficient so far (personal communication, Paul Bertone, EBI)

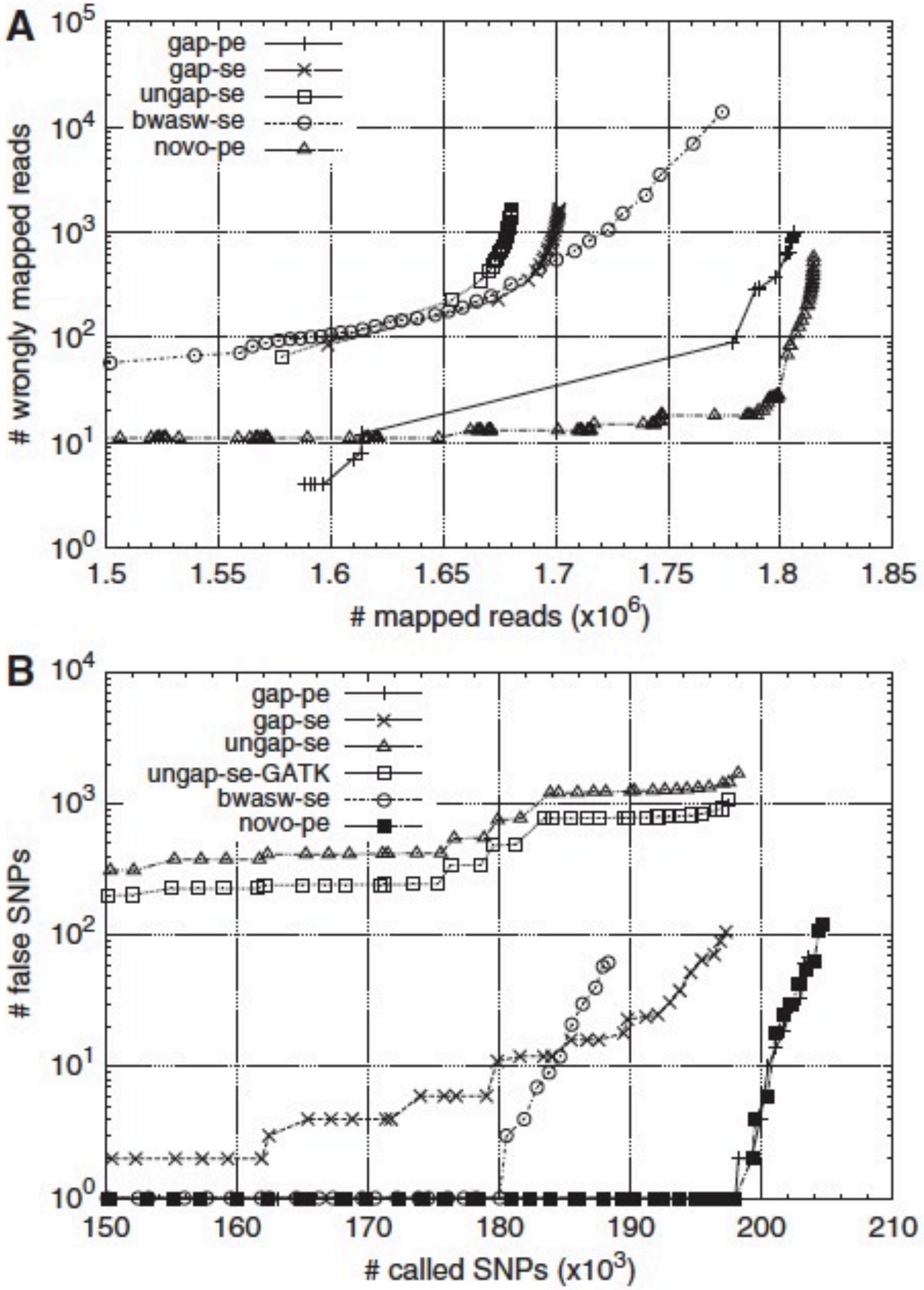
Recent developments

- gapped alignment
    - Recent aligners are able to perform gapped alignments
        - small indels
        - no splicing events with large introns
    - BWA, Novoalign

- bisulfite sequencing
    - unmethylated C are converted to T (G complement converted to A)
    - 2 references
        - one with all C converted to T
        - one with all G converted to A
        - C-T mismatch or G-A mismatch are ignored
        - results from both alignments are combined

What aligner for my data?

- The choice of aligner depends on the data at hands (too late!)
- “Early”: it should be decided when planning the experiment
- What criteria?
  - do you always need paired end reads?
  - do you need gap alignments?

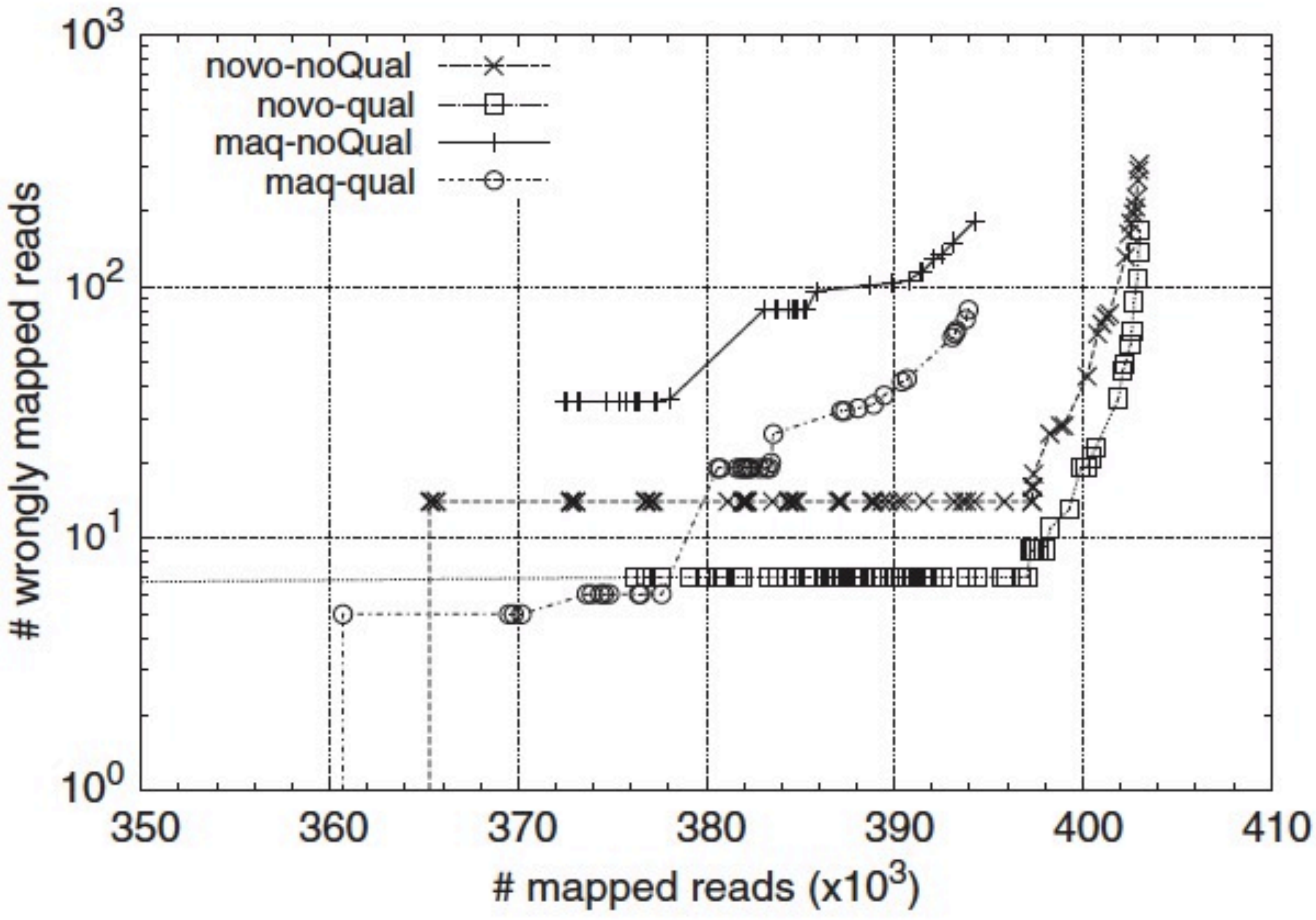


EMBL

Using read quality

- lower penalty for base with lower qualities

- quality recalibration helps



Alignment usage summary

- gapped alignment for very short reads (25-36bp) is computationally challenging
  - gapped align. have a better sensitivity, same error rate
  - important for indels and SNPs
  - impact not analyzed for ChIP-Seq or RNA-Seq
- paired end alignment always outperform single end alignment

- Next tools to come:
  - multi-genome alignment (1000 genomes project, Drosophila population genomics project, 1001 genomes project...)

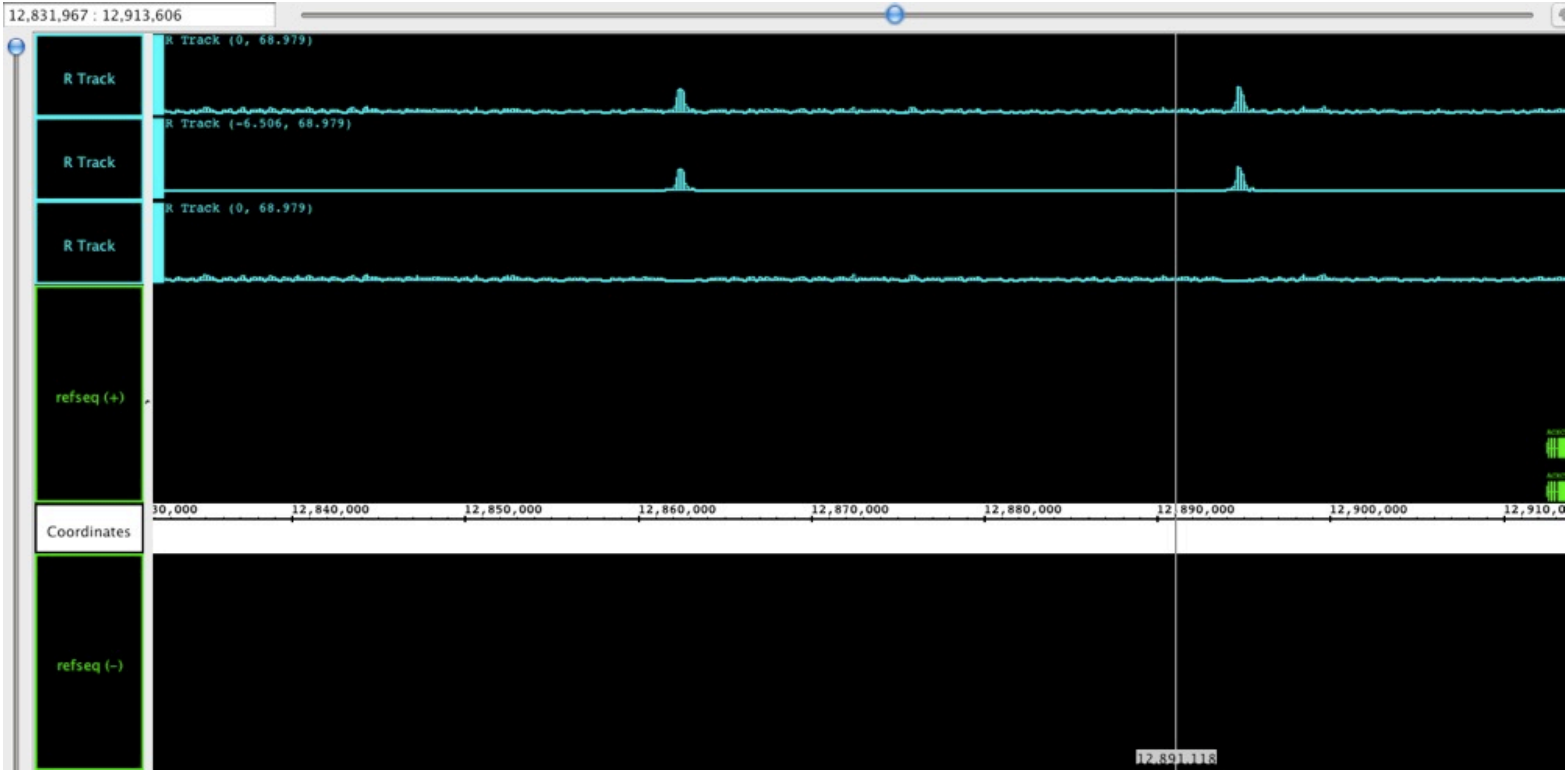
Aligner’s usage, an example

- What is the impact of unique alignments?

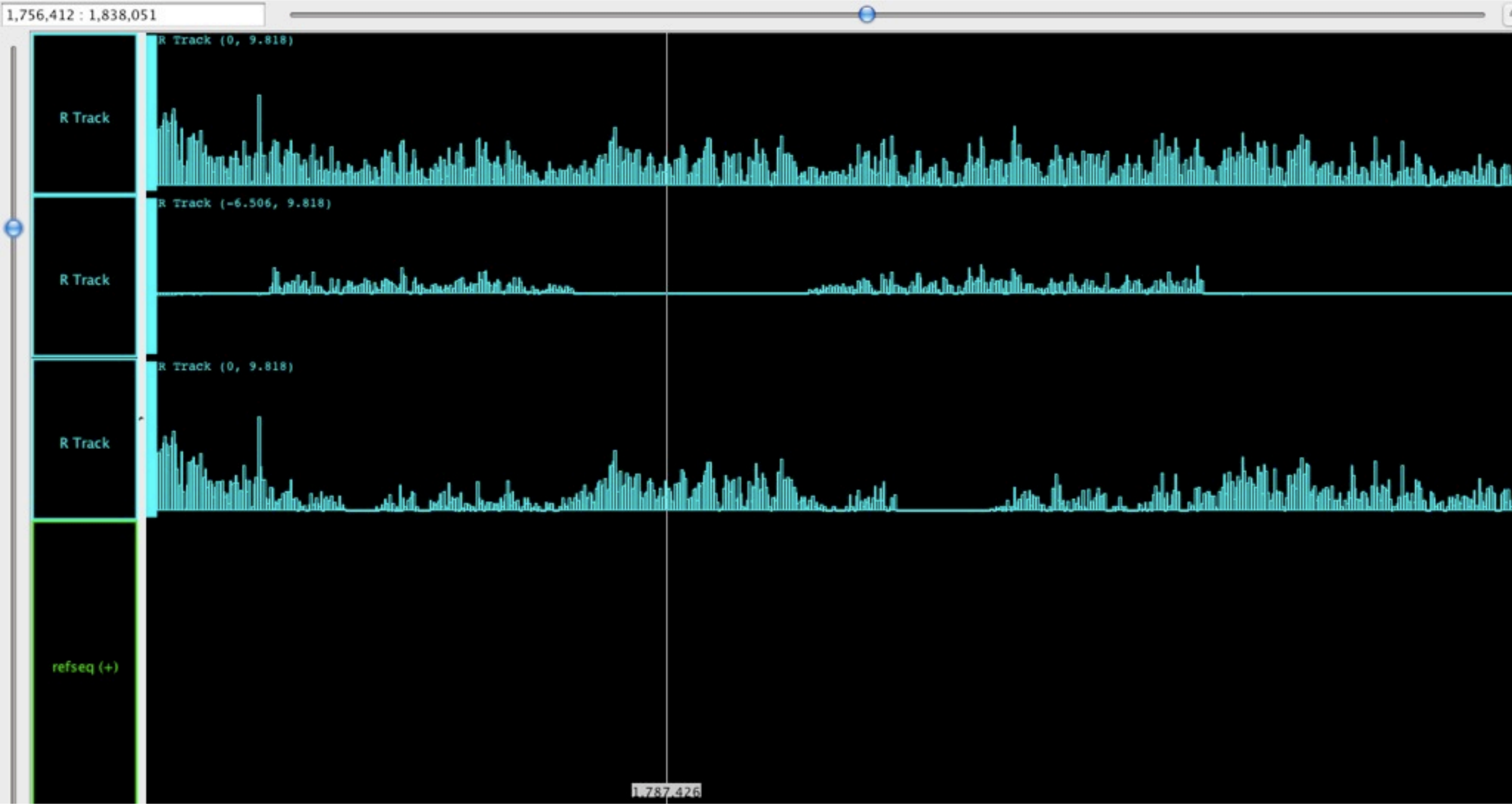
- Approach:
    - MAQ policy: keep one alignment per read
    - strict policy: keep only reads with a single alignment

- How to assess the differences?
    - comparing MAQ, strict and (MAQ - strict)

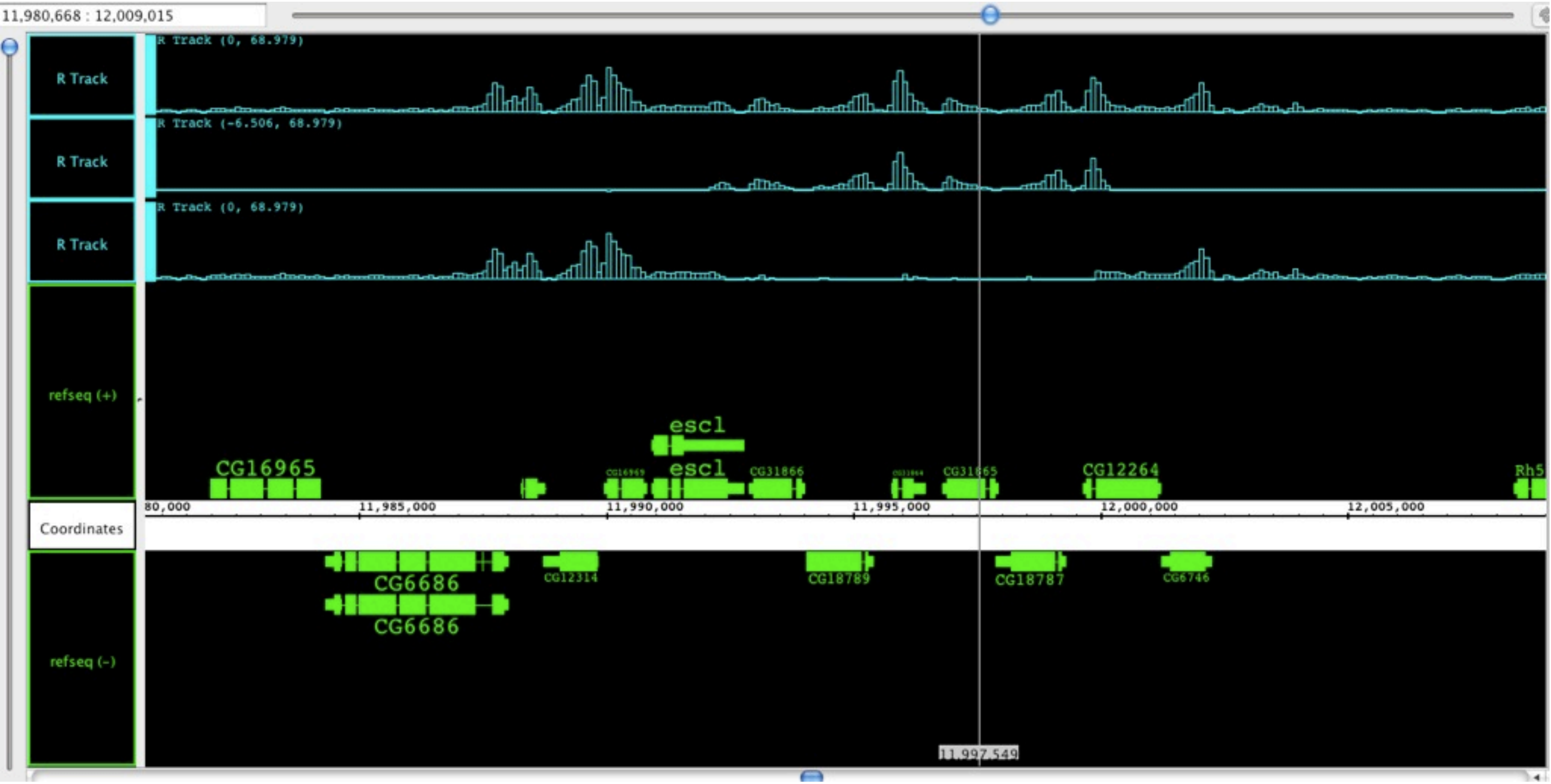
- Data
    - ChIP-Seq of an histone mark: K27Ac



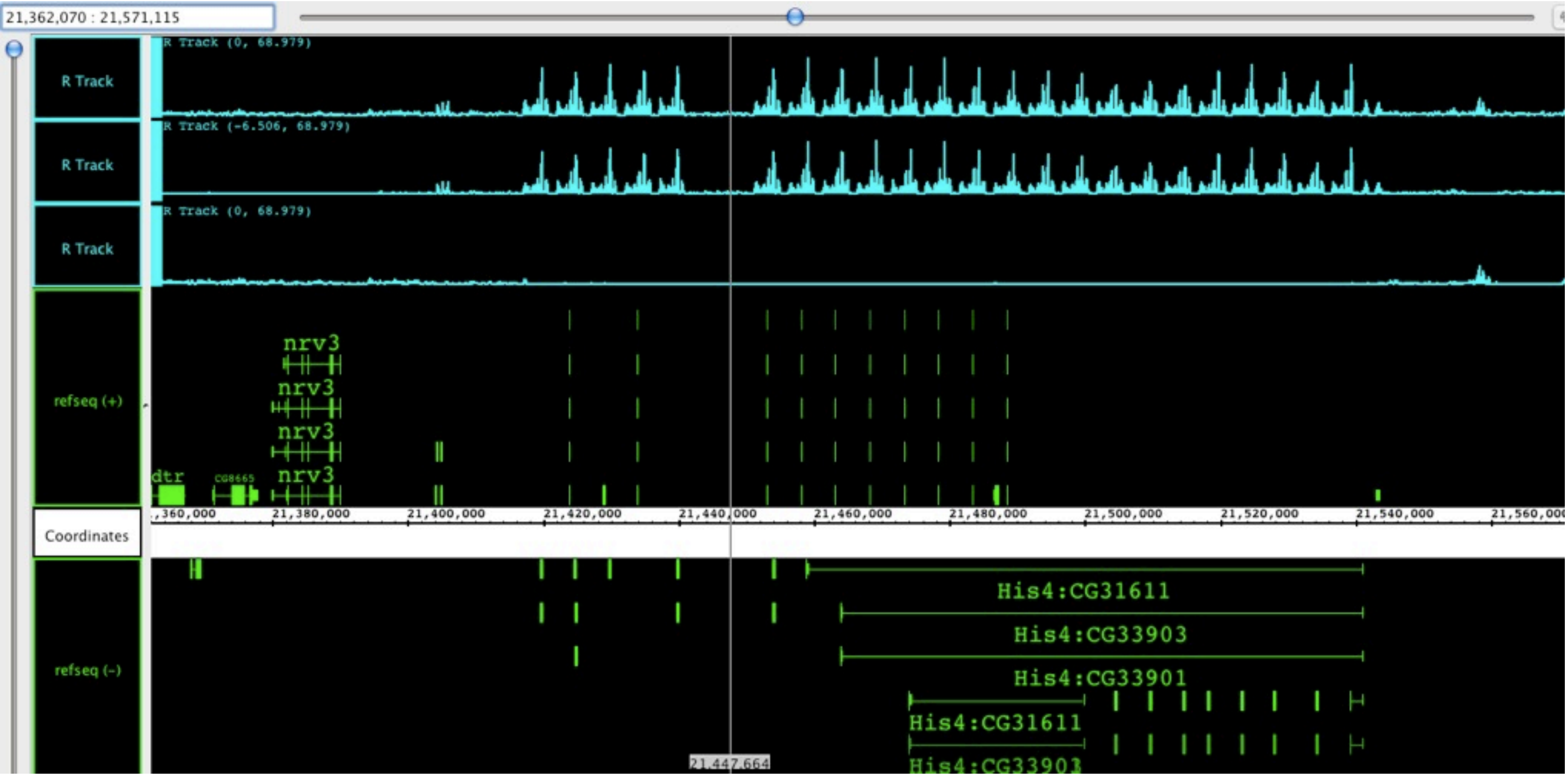
Most are harmless: repetitive region small


1,756,412 : 1,838,051
R Track
R Track
R Track
refseq (+)
R Track (0, 9.818)
R Track (-6.506, 9.818)
R Track (0, 9.818)
1,787,426

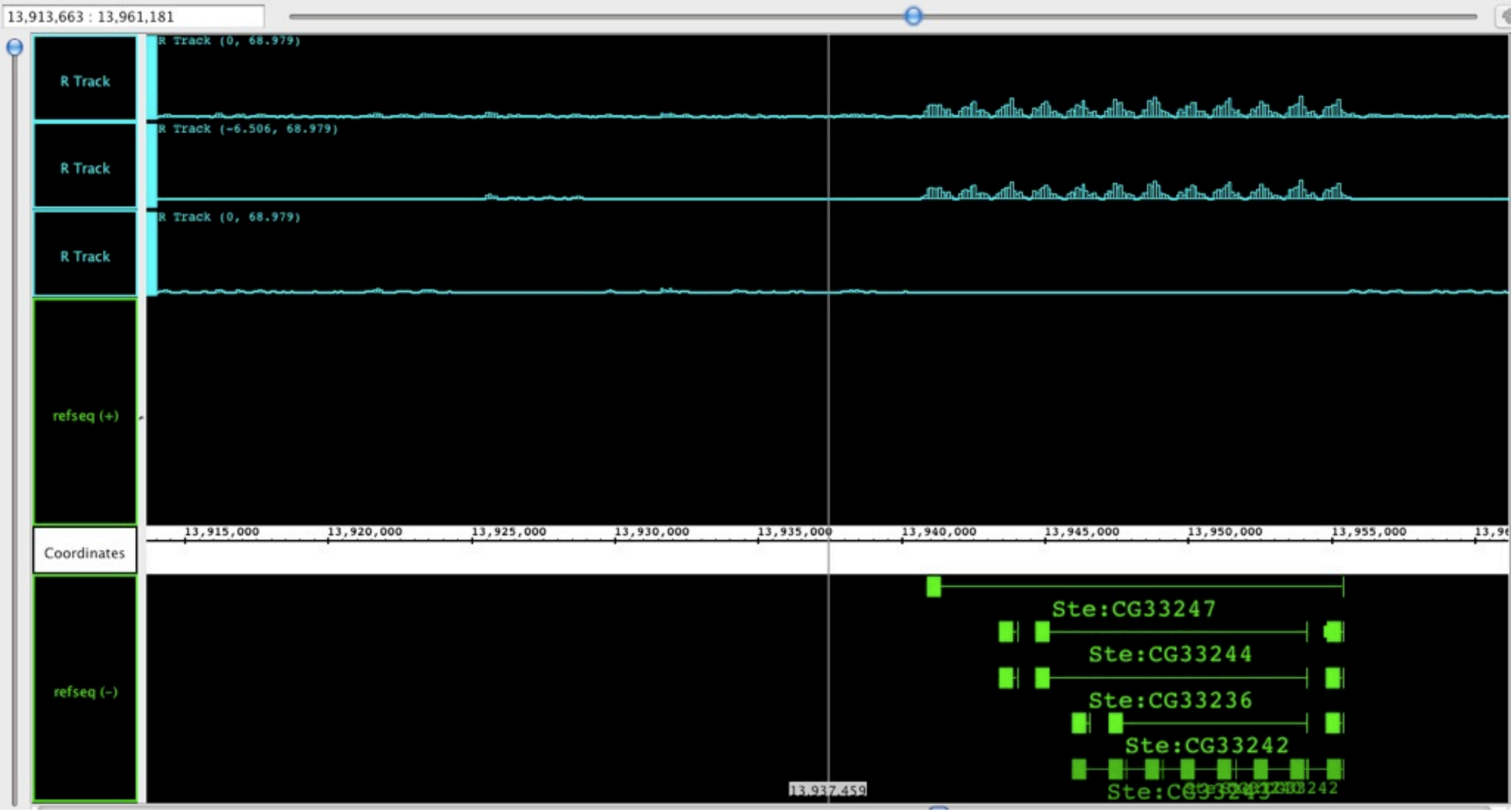
or wide



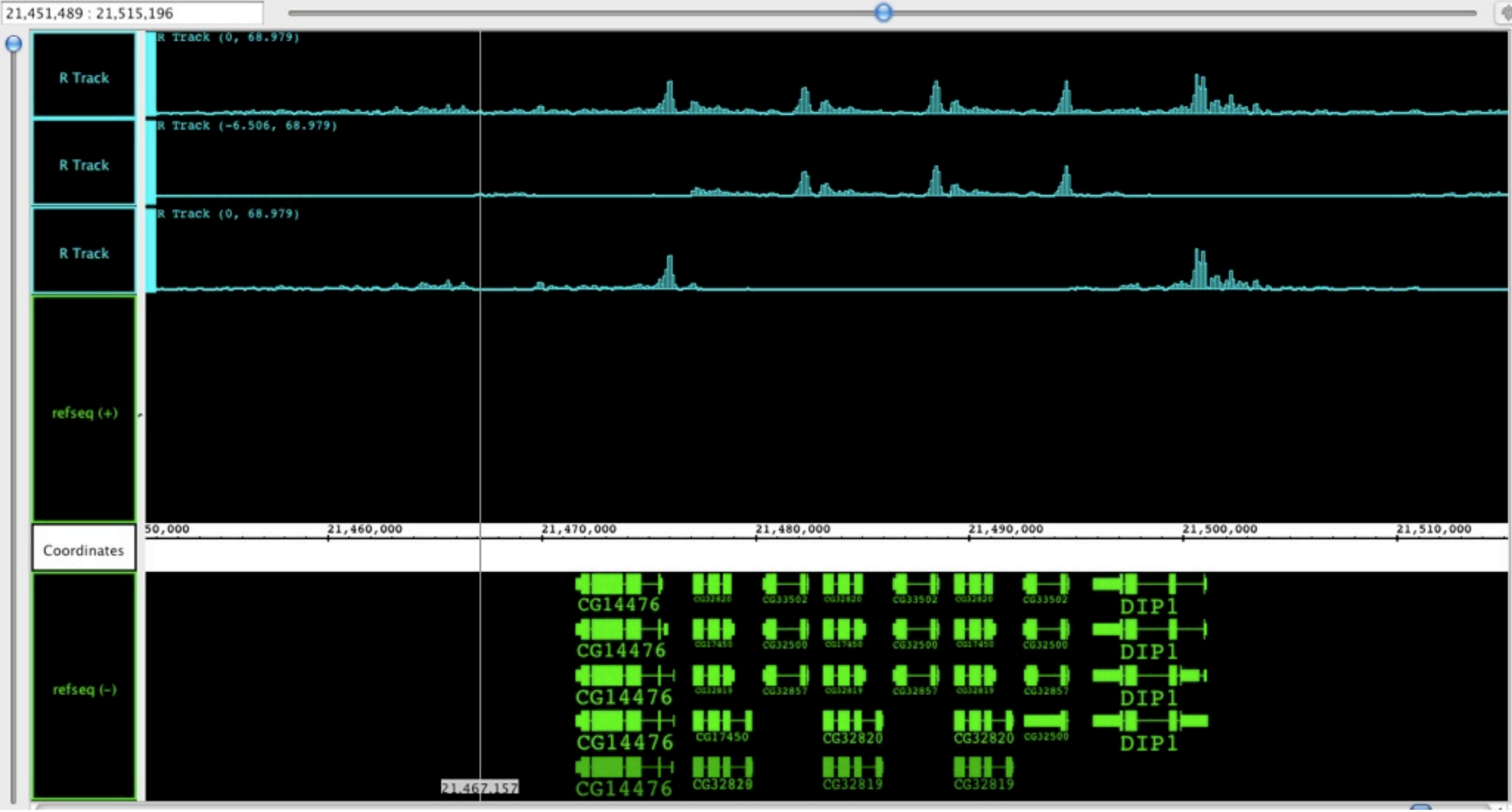
Few result in loss of information



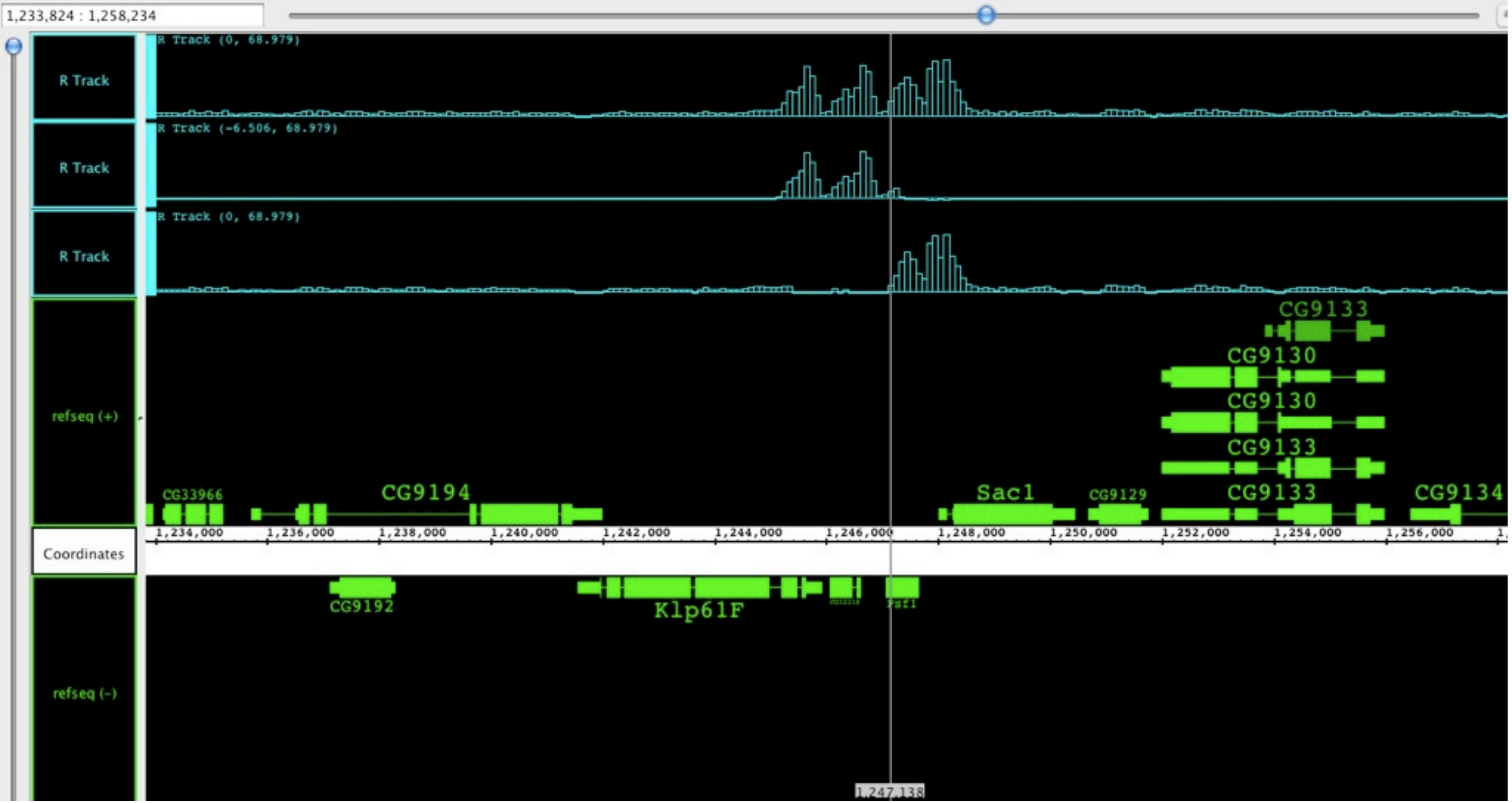
Most of these are very repeated elements: Histone cluster



Protein kinase involved in spermatogenesis


21,451,489 : 21,515,196
R Track
R Track (0, 68.979)
R Track
R Track (-6.506, 68.979)
R Track
R Track (0, 68.979)
refseq (+)
Coordinates
50,000
21,460,000
21,470,000
21,480,000
21,490,000
21,500,000
21,510,000
refseq (-)
CG14476
CG32820
CG33502
CG32820
CG33502
CG32820
CG33502
DIP1
CG14476
CG17450
CG32500
CG17450
CG32500
CG17450
CG32500
DIP1
CG14476
CG32819
CG32857
CG32819
CG32857
CG32819
CG32857
DIP1
CG14476
CG17450
CG32820
CG32820
CG32500
DIP1
21,467,157
CG14476
CG32820
CG32819
CG32819

or unknowm...



Extremely few are not clusters.

Unique alignment summary

- Always important to assess the aligner's effect as every aligner introduces technical biases!

- In that example, using the strict policy should
  - simplify the peak calling
  - reduces the false positives in downstream analyses
  - has only a few side-effects (redo with a gene mark?)

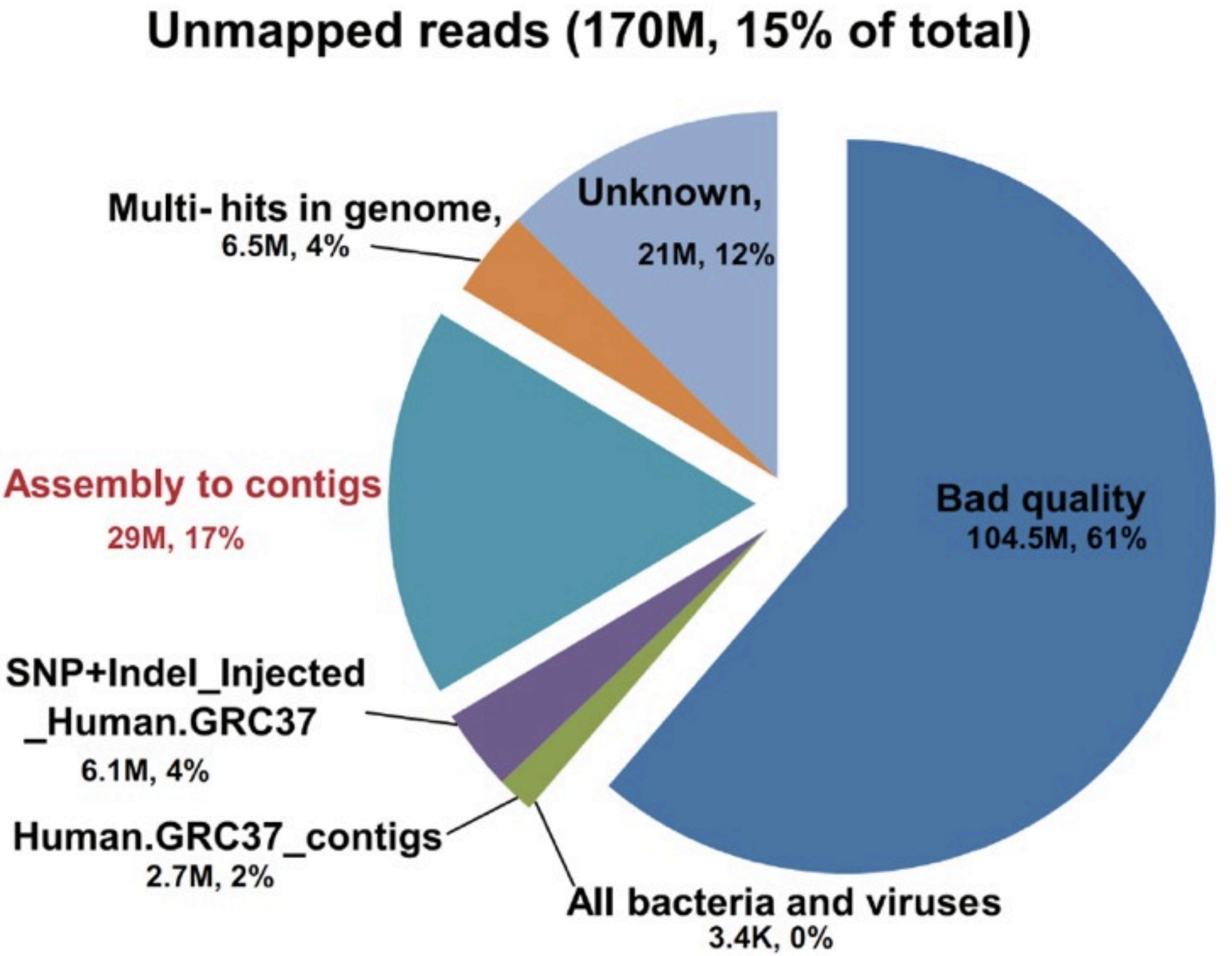
- Additional information to be extracted and used downstream
  - For visualization, use a mappability track
  - Filter the annotation not to introduce false negatives in the analyzes

Another caveat: what reference?

- How close is your sample’s genome to the published available reference one?

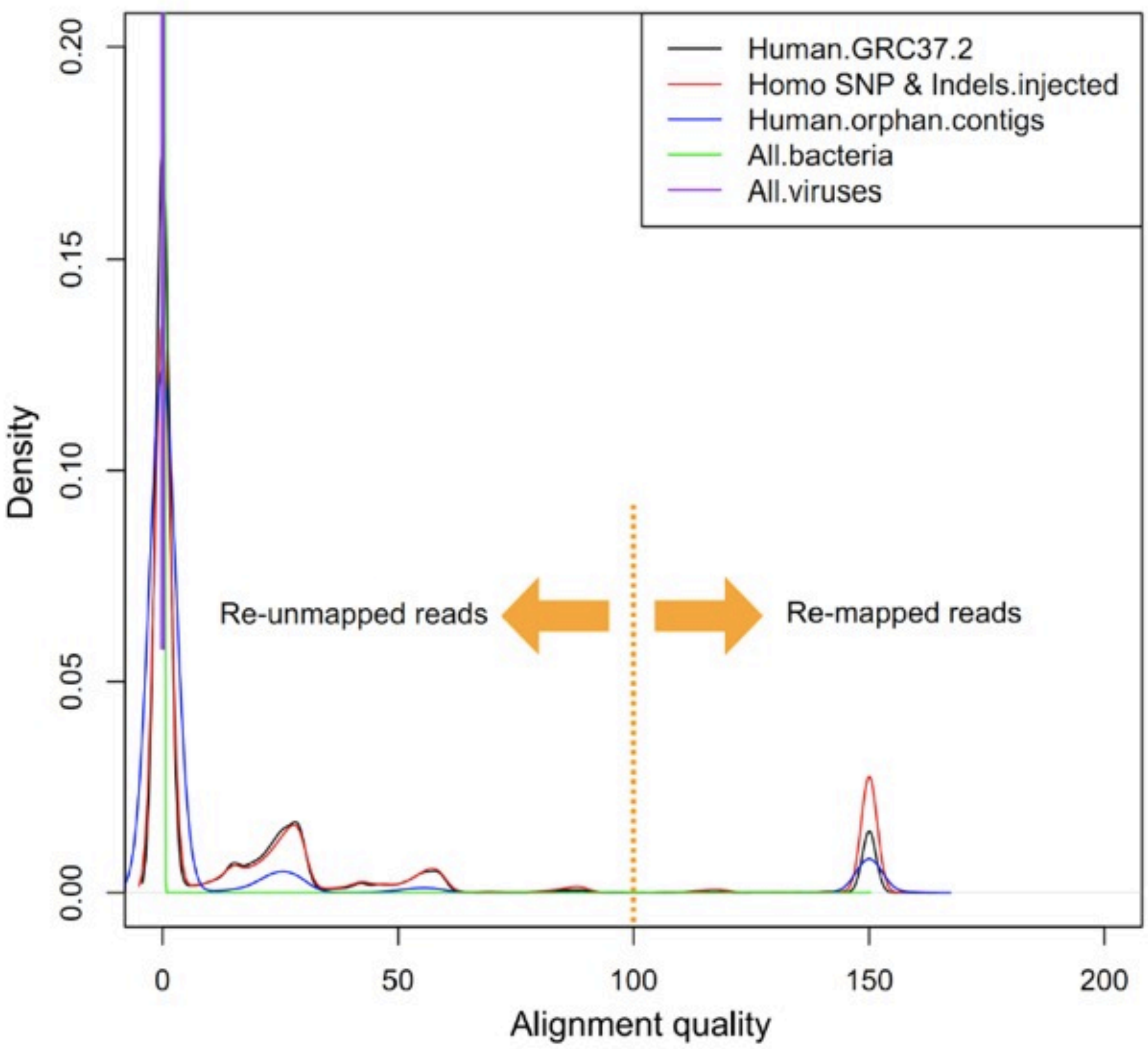
- Specific kind of data, such as RNA-Seq:
  - genome or transcriptome?
  - what about novel exon-exon junctions?

Reference modification



Personalized reference

- Identify SNPs and indels
- Inject them into the "reference" genome
- A "personalized" genome that rescues "only" ~4% of unmapped reads
- but significantly reduces false positive SNPs



Technical artifact or amazing new biology?

- A recent paper that spills a lot of taint:
  - Li et al. Widespread RNA and DNA Sequence Differences in the Human Transcriptome. Science (2011)
  - Major critics (Joe Pickrell):
    - http://www.genomesunzipped.org/2011/05/notes-on-the-evidence-for-extensive-rna-editing-in-humans.php

What they did

- Compare RNA and DNA from matched samples
    - observe numerous events where RNA != DNA
    - process known as RNA editing
    - known in human:
        - an enzyme convert A into I (Inosine) recognized as a G during translation
        - another less frequently observed event frmo another enzyme:
            - C -> U
- BUT they observe all possible conversions!

What might be

- They use reads aligning uniquely to the genome.

- The main point can be summarized like this: RNA editing involves the production of two different RNA and/or protein sequences from a single DNA sequence. To infer RNA editing from the presence of two different RNA and/or protein sequences, then, one must be very sure that they derive from the same DNA sequence, rather than from two different copies of the DNA (due to, for example, paralogs or copy number variants).

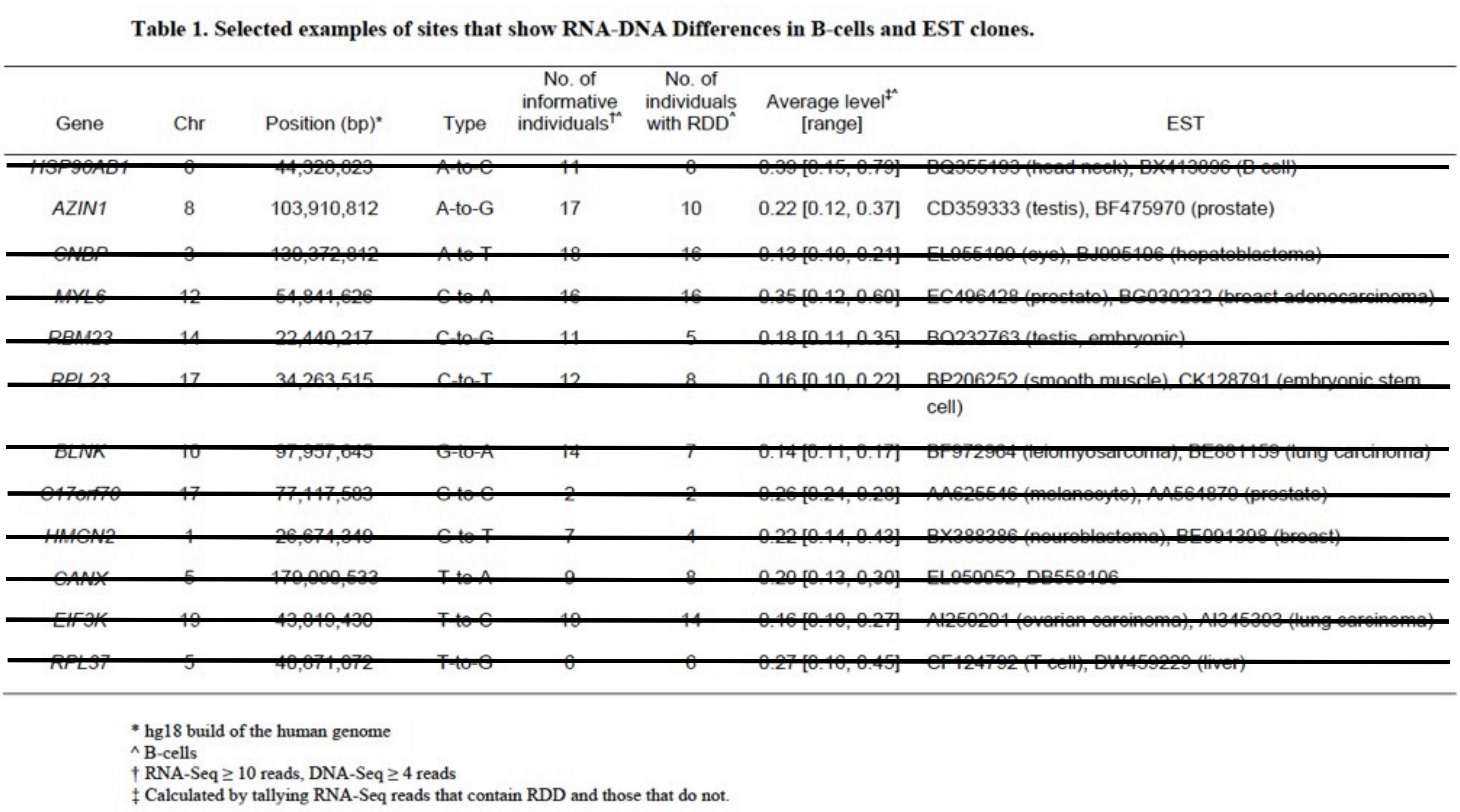


Table 1. Selected examples of sites that show RNA-DNA Differences in B-cells and EST clones.

Gene	Chr	Position (bp)[*]	Type	No. of informative individuals[†^]	No. of individuals with RDD[^]	Average level[‡^] [range]	EST
~~*HSP90AB1*~~	~~6~~	~~44,328,823~~	~~A-to-C~~	~~11~~	~~8~~	~~0.39 [0.15, 0.79]~~	~~BQ355193 (head neck), BX413896 (B cell)~~
AZIN1	8	103,910,812	A-to-G	17	10	0.22 [0.12, 0.37]	CD359333 (testis), BF475970 (prostate)
~~*CNBP*~~	~~3~~	~~130,372,812~~	~~A-to-T~~	~~18~~	~~16~~	~~0.13 [0.10, 0.21]~~	~~EL955100 (eye), BJ995106 (hepatoblastoma)~~
~~*MYL6*~~	~~12~~	~~54,841,626~~	~~C-to-A~~	~~16~~	~~16~~	~~0.35 [0.12, 0.60]~~	~~EC496428 (prostate), BG030232 (breast adenocarcinoma)~~
~~*RBM23*~~	~~14~~	~~22,440,217~~	~~C-to-G~~	~~11~~	~~5~~	~~0.18 [0.11, 0.35]~~	~~BQ232763 (testis, embryonic)~~
~~*RPL23*~~	~~17~~	~~34,263,515~~	~~C-to-T~~	~~12~~	~~8~~	~~0.16 [0.10, 0.22]~~	~~BP206252 (smooth muscle), CK128791 (embryonic stem~~ cell)
~~*BLNK*~~	~~10~~	~~97,957,645~~	~~G-to-A~~	~~14~~	~~7~~	~~0.14 [0.11, 0.17]~~	~~BF972964 (leiomyosarcoma), BE881159 (lung carcinoma)~~
~~*C17orf70*~~	~~17~~	~~77,117,583~~	~~G-to-C~~	~~2~~	~~2~~	~~0.26 [0.24, 0.28]~~	~~AA625546 (melanocyte), AA564870 (prostate)~~
~~*HMGN2*~~	~~1~~	~~26,674,340~~	~~G-to-T~~	~~7~~	~~4~~	~~0.22 [0.14, 0.43]~~	~~BX388386 (neuroblastoma), BE001398 (breast)~~
~~*CANX*~~	~~5~~	~~179,090,533~~	~~T-to-A~~	~~9~~	~~8~~	~~0.20 [0.13, 0.30]~~	~~EL950052, DB558106~~
~~*EIF3K*~~	~~19~~	~~43,819,430~~	~~T-to-C~~	~~19~~	~~14~~	~~0.16 [0.10, 0.27]~~	~~AI250201 (ovarian carcinoma), AI345303 (lung carcinoma)~~
~~*RPL37*~~	~~5~~	~~40,871,872~~	~~T-to-G~~	~~6~~	~~6~~	~~0.27 [0.16, 0.45]~~	~~CF124792 (T cell), DW459229 (liver)~~

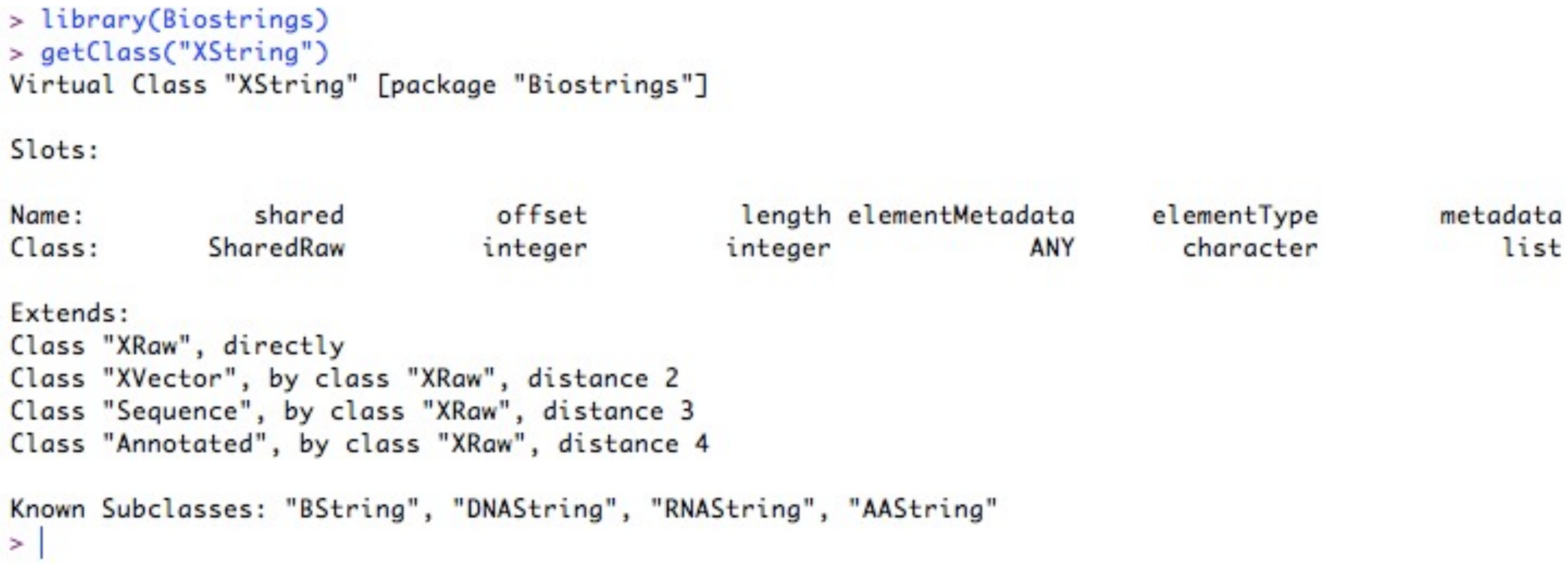
[*] hg18 build of the human genome
[^] B-cells
[†] RNA-Seq ≥ 10 reads, DNA-Seq ≥ 4 reads
[‡] Calculated by tallying RNA-Seq reads that contain RDD and those that do not.

More pleasant news

- Bioconductor offers many new possibilities including:
  - pattern matching,
  - pairwise alignment,
  - SNPs injection
  - ...

The Biostrings package

- All the classes in that package derives from the *XString* class



```
> library(Biostrings)
> getClass("XString")
Virtual Class "XString" [package "Biostrings"]

Slots:

Name:           shared          offset          length elementMetadata     elementType        metadata
Class:       SharedRaw         integer         integer             ANY       character            list

Extends:
Class "XRaw", directly
Class "XVector", by class "XRaw", distance 2
Class "Sequence", by class "XRaw", distance 3
Class "Annotated", by class "XRaw", distance 4

Known Subclasses: "BString", "DNAString", "RNAString", "AAString"
> |
```

- There are 4 subclasses:
  - *BString*: store strings without alphabet
  - *DNAString*: store strings with an DNA alphabet
  - *RNAString*: store strings with an RNA alphabet
  - *AAString*: store strings with an Amino Acid alphabet

XString Methods

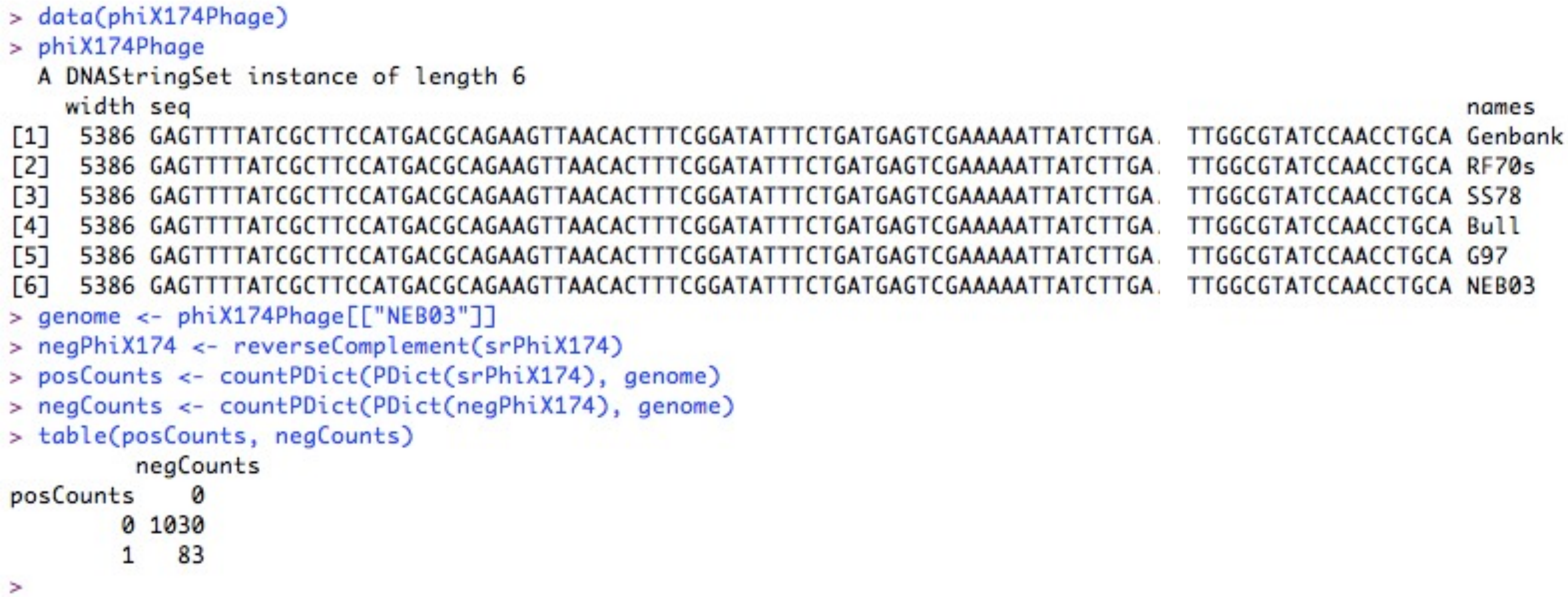
- Basic utilities
  - subsequence selection
    - subseq, Views, narrow (XStringSet, IRanges package)
  - letter frequencies
    - alphabetFrequency, *di*nucleotideFrequency (*tri..., oligo...*), uniqueLetters
  - letter consensus
    - consensusMatrix, consensusString
  - letter transformation
    - reverse, complement, reverseComplement, translate, chartr
  - Input / Output
    - read.*DNA*StringSet (*...B..., ...RNA..., ...AA...*)
    - write.XStringSet, save.XStringSet

XString Methods (c’ed)

- Advanced:
  - alignment utilities
    - pairwiseAlignment, stringDist
  - string matching
    - matchPDict (on a reference or a reference set (v))
      - (v)matchPDict, (v)countPDict, (v)whichPDict
    - matchPattern
      - (v)matchPattern, (v)countPattern, neditStartingAt, neditEndingAt, (which.)isMatchingStartingAt, (which.)isMatchingEndingAt
    - matchPWM
      - matchPWM, countPWM
  - others
    - matchLRPatterns, trimLRPatterns, matchProbePair, findPalindromes, findComplementedPalindromes

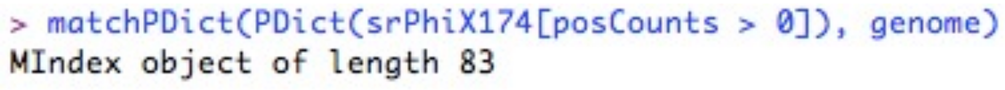
Example 4: String Matching

- Match counting



```
> data(phiX174Phage)
> phiX174Phage
  A DNAStringSet instance of length 6
    width seq                                                                                   names
[1]  5386 GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAAATTATCTTGA.  TTGGCGTATCCAACCTGCA Genbank
[2]  5386 GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAAATTATCTTGA.  TTGGCGTATCCAACCTGCA RF70s
[3]  5386 GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAAATTATCTTGA.  TTGGCGTATCCAACCTGCA SS78
[4]  5386 GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAAATTATCTTGA.  TTGGCGTATCCAACCTGCA Bull
[5]  5386 GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAAATTATCTTGA.  TTGGCGTATCCAACCTGCA G97
[6]  5386 GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAAATTATCTTGA.  TTGGCGTATCCAACCTGCA NEB03
> genome <- phiX174Phage[["NEB03"]]
> negPhiX174 <- reverseComplement(srPhiX174)
> posCounts <- countPDict(PDict(srPhiX174), genome)
> negCounts <- countPDict(PDict(negPhiX174), genome)
> table(posCounts, negCounts)
         negCounts
posCounts    0
        0 1030
        1   83
>
```

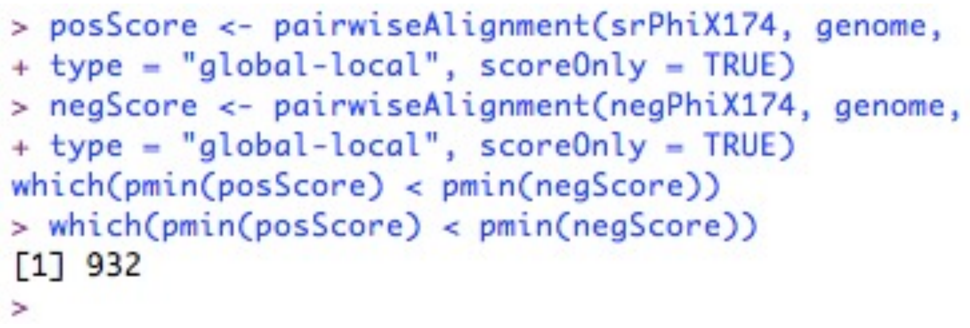
- So we have 1030 reads that do not align either way to the genome and only 83 aligning (and don't ask me why...).
- The match locations can be found using:



```
> matchPDict(PDict(srPhiX174[posCounts > 0]), genome)
MIndex object of length 83
```

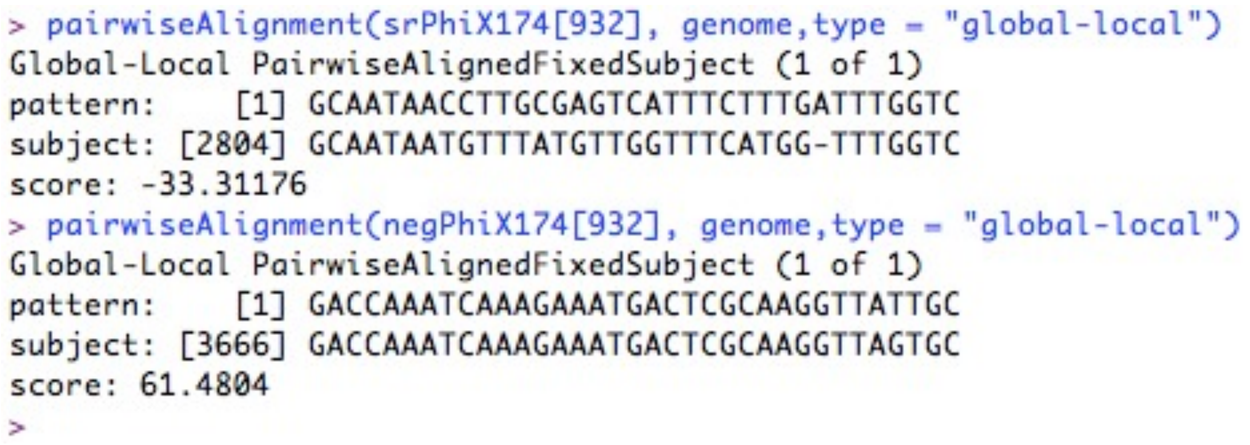
Example 5: Pairwise alignment

- alignment scores



```
> posScore <- pairwiseAlignment(srPhiX174, genome,
+ type = "global-local", scoreOnly = TRUE)
> negScore <- pairwiseAlignment(negPhiX174, genome,
+ type = "global-local", scoreOnly = TRUE)
which(pmin(posScore) < pmin(negScore))
> which(pmin(posScore) < pmin(negScore))
[1] 932
>
```

- alignment



```
> pairwiseAlignment(srPhiX174[932], genome,type = "global-local")
Global-Local PairwiseAlignedFixedSubject (1 of 1)
pattern:    [1] GCAATAACCTTGCGAGTCATTTCTTTGATTTGGTC
subject: [2804] GCAATAATGTTTATGTTGGTTTCATGG-TTTGGTC
score: -33.31176
> pairwiseAlignment(negPhiX174[932], genome,type = "global-local")
Global-Local PairwiseAlignedFixedSubject (1 of 1)
pattern:    [1] GACCAAATCAAAGAAATGACTCGCAAGGTTATTGC
subject: [3666] GACCAAATCAAAGAAATGACTCGCAAGGTTAGTGC
score: 61.4804
>
```

The next level

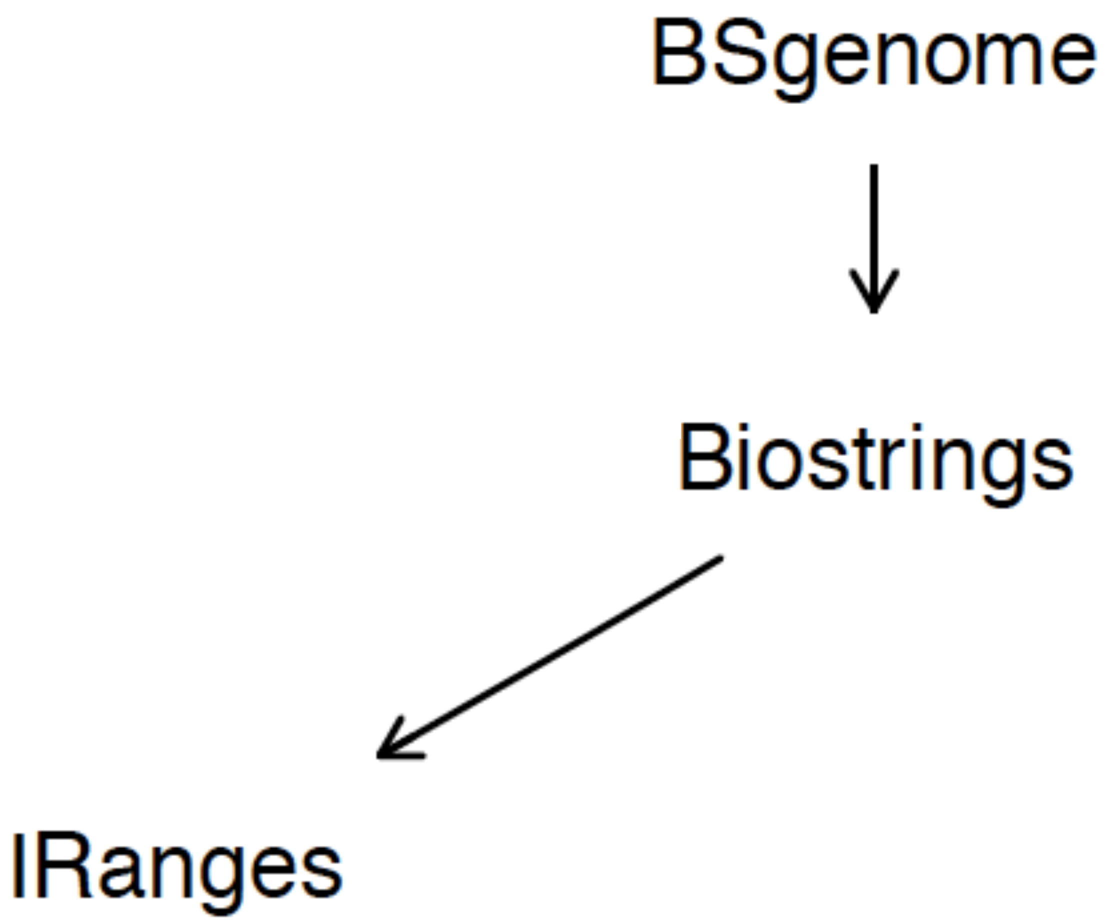
- Biostrings offers tools to deal with biologically meaningful intervals and objects.

- Many organism have been sequenced and their genome is known.

- An interface in R to easily access and manipulate such information: the **BSgenome** package.

BSgenome

- It is not just a data package; it leverages the functionalities introduced in **Biostrings**.

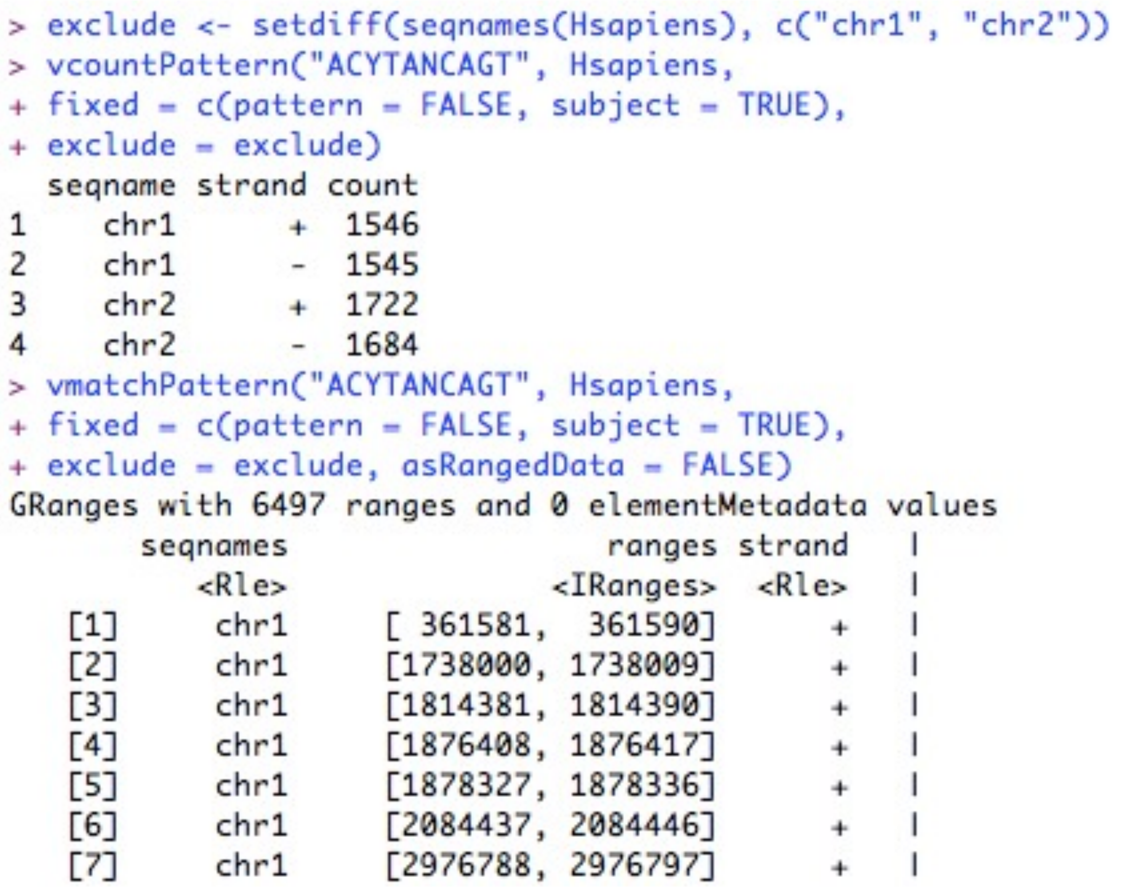


BSgenome methods

- Sequence selection
  - [[, $
- Subsequence selection
  - getSeq
- Accessors
  - length,names/seqnames, mseqnames, seqlengths, masknames, sourceUrl
- Matching
  - all Biostrings methods
- SNPs
  - injectSNPs, SNPlocs_pkgname, SNPcount, SNPlocs

Extending Biostrings: example 1

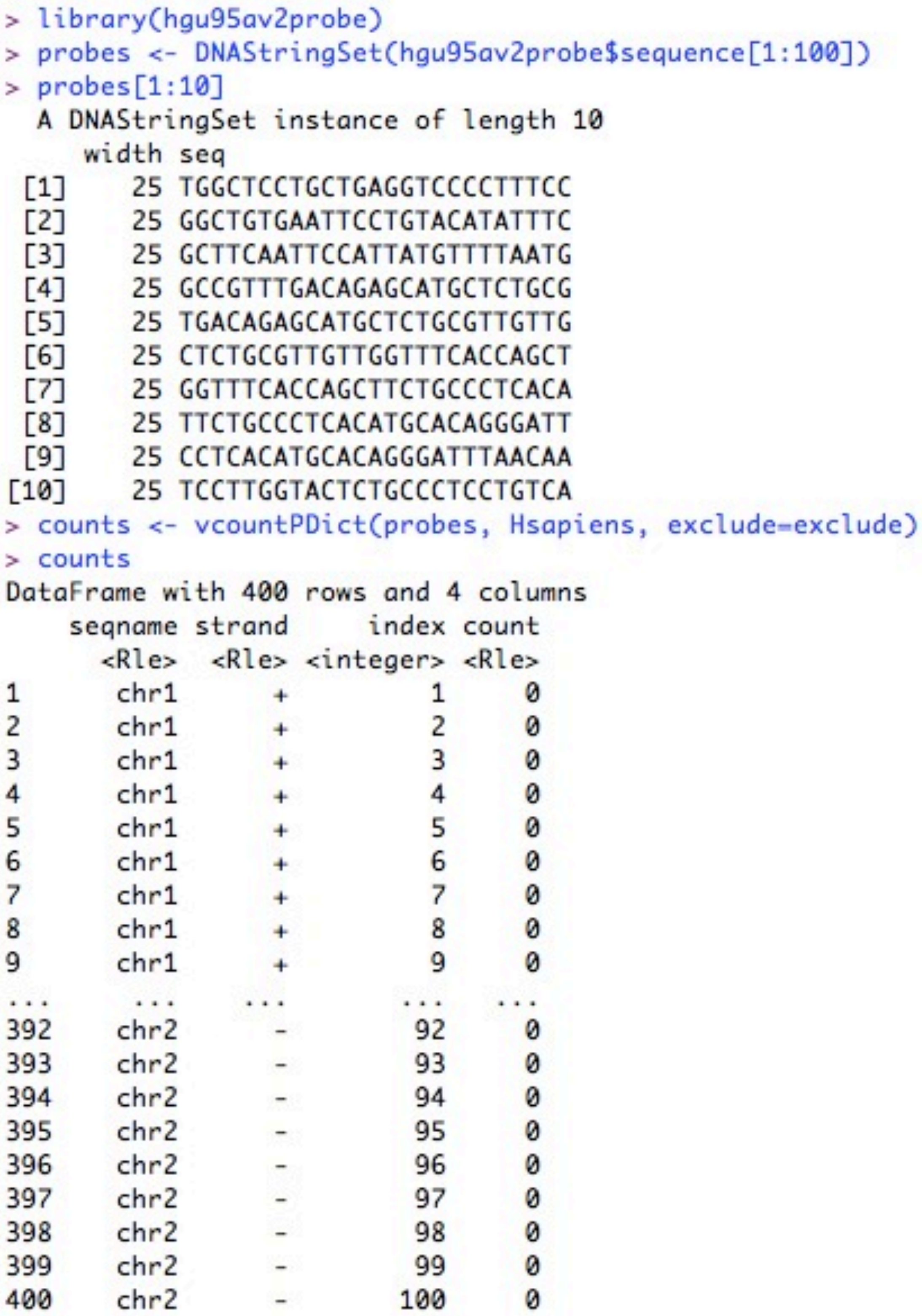
- Applying the Biostrings matching functions:



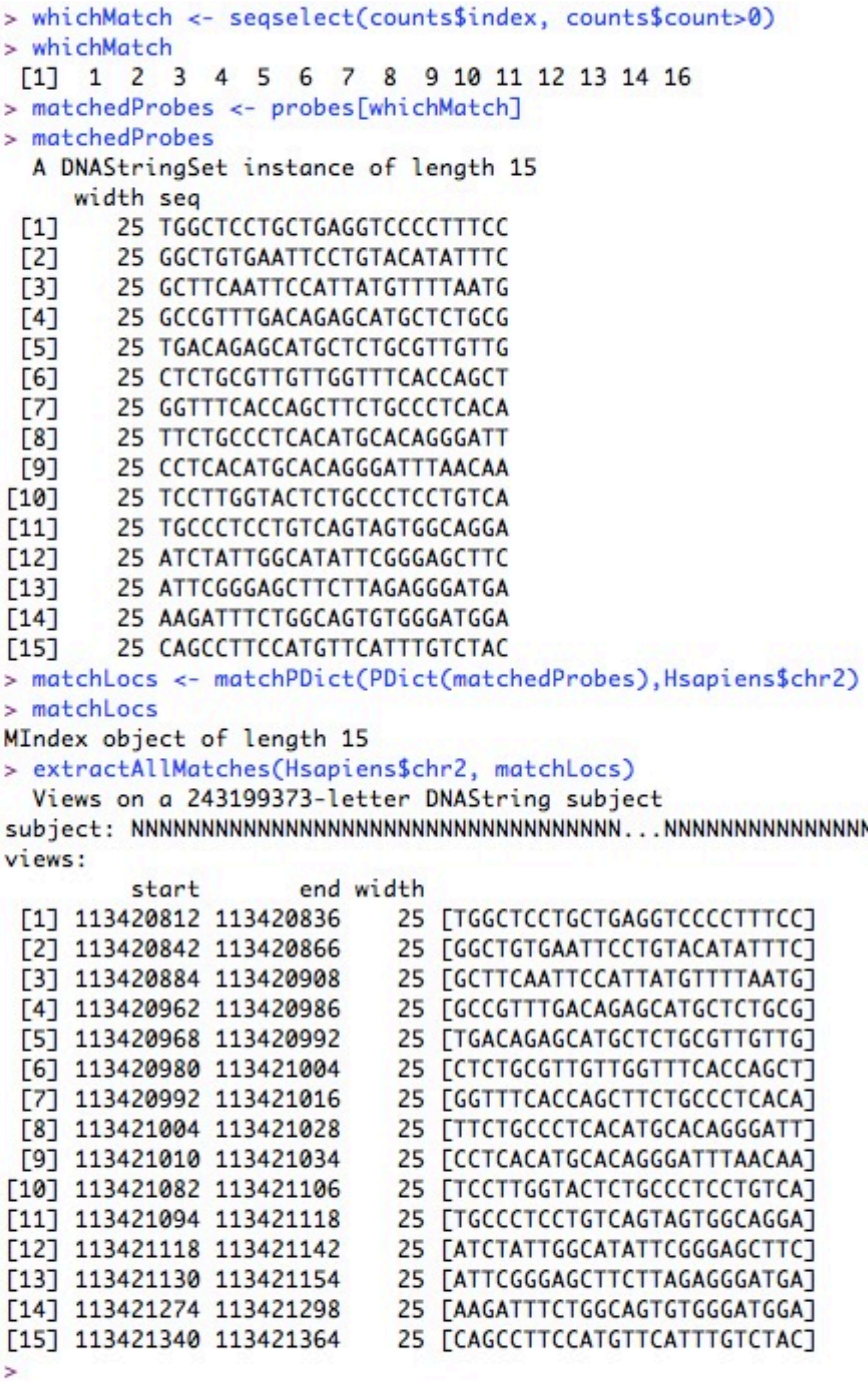
```
> exclude <- setdiff(seqnames(Hsapiens), c("chr1", "chr2"))
> vcountPattern("ACYTANCAGT", Hsapiens,
+ fixed = c(pattern = FALSE, subject = TRUE),
+ exclude = exclude)
  seqname strand count
1    chr1      +  1546
2    chr1      -  1545
3    chr2      +  1722
4    chr2      -  1684
> vmatchPattern("ACYTANCAGT", Hsapiens,
+ fixed = c(pattern = FALSE, subject = TRUE),
+ exclude = exclude, asRangedData = FALSE)
GRanges with 6497 ranges and 0 elementMetadata values
       seqnames                 ranges strand   |
          <Rle>              <IRanges>  <Rle>   |
   [1]     chr1     [ 361581,  361590]      +   |
   [2]     chr1     [1738000, 1738009]      +   |
   [3]     chr1     [1814381, 1814390]      +   |
   [4]     chr1     [1876408, 1876417]      +   |
   [5]     chr1     [1878327, 1878336]      +   |
   [6]     chr1     [2084437, 2084446]      +   |
   [7]     chr1     [2976788, 2976797]      +   |
```

Example 2

- Using a Pattern Dictionary, e.g. a library of microarray probes



```
> library(hgu95av2probe)
> probes <- DNAStringSet(hgu95av2probe$sequence[1:100])
> probes[1:10]
  A DNAStringSet instance of length 10
     width seq
 [1]    25 TGGCTCCTGCTGAGGTCCCCTTTCC
 [2]    25 GGCTGTGAATTCCTGTACATATTTC
 [3]    25 GCTTCAATTCCATTATGTTTTAATG
 [4]    25 GCCGTTTGACAGAGCATGCTCTGCG
 [5]    25 TGACAGAGCATGCTCTGCGTTGTTG
 [6]    25 CTCTGCGTTGTTGGTTTCACCAGCT
 [7]    25 GGTTTCACCAGCTTCTGCCCTCACA
 [8]    25 TTCTGCCCTCACATGCACAGGGATT
 [9]    25 CCTCACATGCACAGGGATTTAACAA
[10]    25 TCCTTGGTACTCTGCCCTCCTGTCA
> counts <- vcountPDict(probes, Hsapiens, exclude=exclude)
> counts
DataFrame with 400 rows and 4 columns
    seqname strand     index count
      <Rle>  <Rle> <integer> <Rle>
1      chr1      +         1     0
2      chr1      +         2     0
3      chr1      +         3     0
4      chr1      +         4     0
5      chr1      +         5     0
6      chr1      +         6     0
7      chr1      +         7     0
8      chr1      +         8     0
9      chr1      +         9     0
...     ...    ...       ...   ...
392    chr2      -        92     0
393    chr2      -        93     0
394    chr2      -        94     0
395    chr2      -        95     0
396    chr2      -        96     0
397    chr2      -        97     0
398    chr2      -        98     0
399    chr2      -        99     0
400    chr2      -       100     0
```

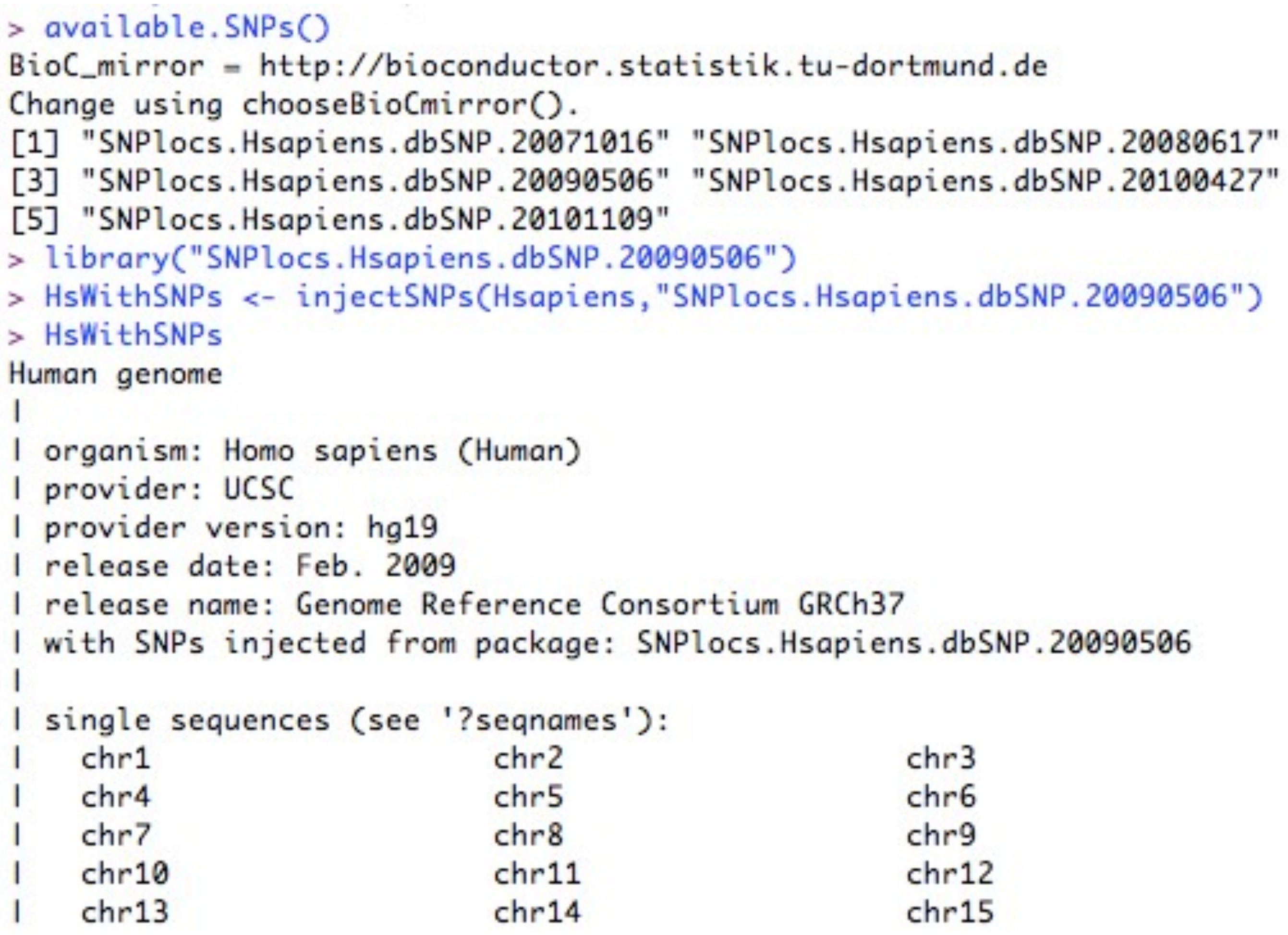


```
> whichMatch <- seqselect(counts$index, counts$count>0)
> whichMatch
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 16
> matchedProbes <- probes[whichMatch]
> matchedProbes
  A DNAStringSet instance of length 15
     width seq
 [1]    25 TGGCTCCTGCTGAGGTCCCCTTTCC
 [2]    25 GGCTGTGAATTCCTGTACATATTTC
 [3]    25 GCTTCAATTCCATTATGTTTTAATG
 [4]    25 GCCGTTTGACAGAGCATGCTCTGCG
 [5]    25 TGACAGAGCATGCTCTGCGTTGTTG
 [6]    25 CTCTGCGTTGTTGGTTTCACCAGCT
 [7]    25 GGTTTCACCAGCTTCTGCCCTCACA
 [8]    25 TTCTGCCCTCACATGCACAGGGATT
 [9]    25 CCTCACATGCACAGGGATTTAACAA
[10]    25 TCCTTGGTACTCTGCCCTCCTGTCA
[11]    25 TGCCCTCCTGTCAGTAGTGGCAGGA
[12]    25 ATCTATTGGCATATTCGGGAGCTTC
[13]    25 ATTCGGGAGCTTCTTAGAGGGATGA
[14]    25 AAGATTTCTGGCAGTGTGGGATGGA
[15]    25 CAGCCTTCCATGTTCATTTGTCTAC
> matchLocs <- matchPDict(PDict(matchedProbes),Hsapiens$chr2)
> matchLocs
MIndex object of length 15
> extractAllMatches(Hsapiens$chr2, matchLocs)
  Views on a 243199373-letter DNAString subject
subject: NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN...NNNNNNNNNNNNNN
views:
          start       end width
 [1] 113420812 113420836    25 [TGGCTCCTGCTGAGGTCCCCTTTCC]
 [2] 113420842 113420866    25 [GGCTGTGAATTCCTGTACATATTTC]
 [3] 113420884 113420908    25 [GCTTCAATTCCATTATGTTTTAATG]
 [4] 113420962 113420986    25 [GCCGTTTGACAGAGCATGCTCTGCG]
 [5] 113420968 113420992    25 [TGACAGAGCATGCTCTGCGTTGTTG]
 [6] 113420980 113421004    25 [CTCTGCGTTGTTGGTTTCACCAGCT]
 [7] 113420992 113421016    25 [GGTTTCACCAGCTTCTGCCCTCACA]
 [8] 113421004 113421028    25 [TTCTGCCCTCACATGCACAGGGATT]
 [9] 113421010 113421034    25 [CCTCACATGCACAGGGATTTAACAA]
[10] 113421082 113421106    25 [TCCTTGGTACTCTGCCCTCCTGTCA]
[11] 113421094 113421118    25 [TGCCCTCCTGTCAGTAGTGGCAGGA]
[12] 113421118 113421142    25 [ATCTATTGGCATATTCGGGAGCTTC]
[13] 113421130 113421154    25 [ATTCGGGAGCTTCTTAGAGGGATGA]
[14] 113421274 113421298    25 [AAGATTTCTGGCAGTGTGGGATGGA]
[15] 113421340 113421364    25 [CAGCCTTCCATGTTCATTTGTCTAC]
>
```

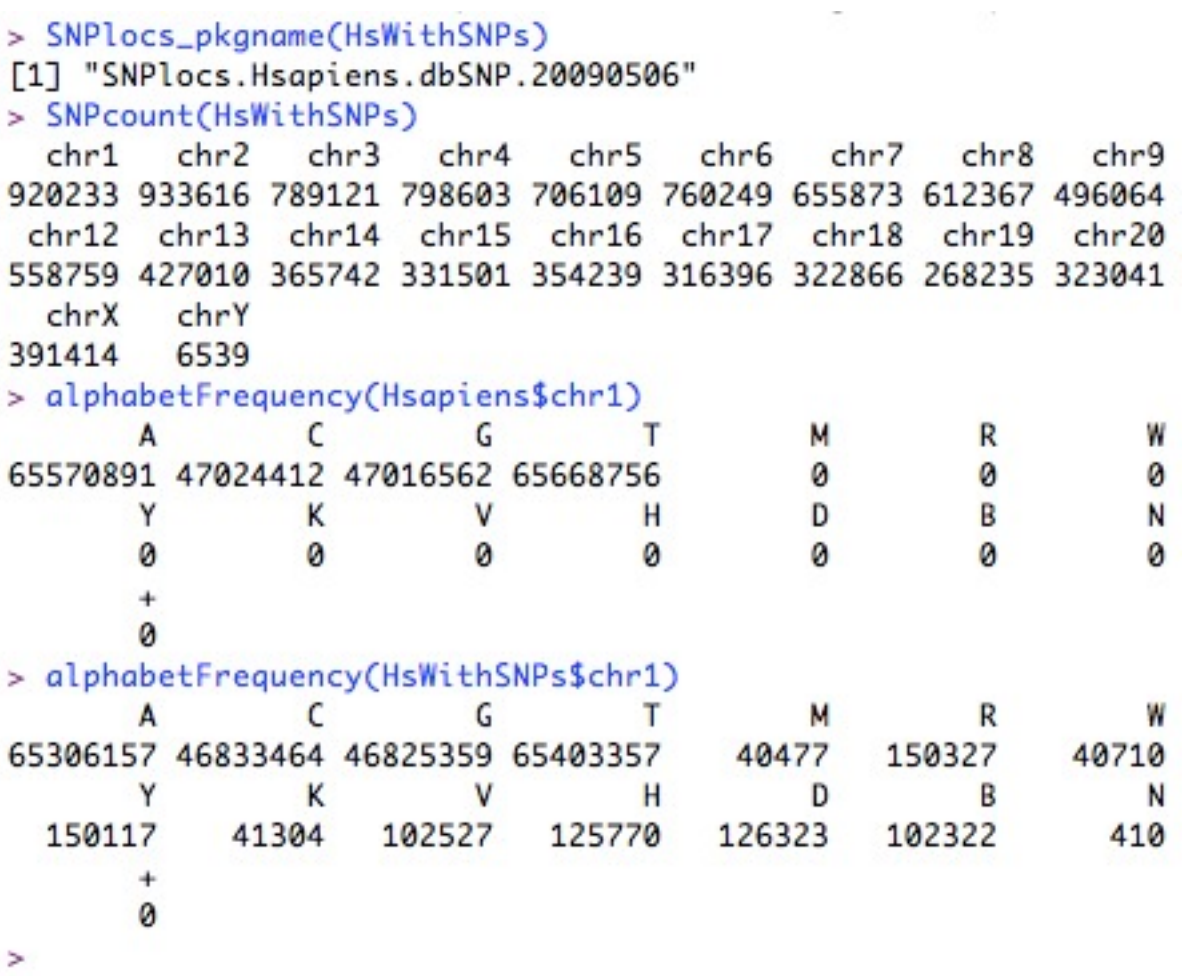
Example 3

- A new interesting feature is the possibility to inject SNPs!

Recent



```
> available.SNPs()
BioC_mirror = http://bioconductor.statistik.tu-dortmund.de
Change using chooseBioCmirror().
[1] "SNPlocs.Hsapiens.dbSNP.20071016" "SNPlocs.Hsapiens.dbSNP.20080617"
[3] "SNPlocs.Hsapiens.dbSNP.20090506" "SNPlocs.Hsapiens.dbSNP.20100427"
[5] "SNPlocs.Hsapiens.dbSNP.20101109"
> library("SNPlocs.Hsapiens.dbSNP.20090506")
> HsWithSNPs <- injectSNPs(Hsapiens,"SNPlocs.Hsapiens.dbSNP.20090506")
> HsWithSNPs
Human genome
|
| organism: Homo sapiens (Human)
| provider: UCSC
| provider version: hg19
| release date: Feb. 2009
| release name: Genome Reference Consortium GRCh37
| with SNPs injected from package: SNPlocs.Hsapiens.dbSNP.20090506
|
| single sequences (see '?seqnames'):
|   chr1                   chr2                   chr3
|   chr4                   chr5                   chr6
|   chr7                   chr8                   chr9
|   chr10                  chr11                  chr12
|   chr13                  chr14                  chr15
```



```
> SNPlocs_pkgname(HsWithSNPs)
[1] "SNPlocs.Hsapiens.dbSNP.20090506"
> SNPcount(HsWithSNPs)
  chr1   chr2   chr3   chr4   chr5   chr6   chr7   chr8   chr9
920233 933616 789121 798603 706109 760249 655873 612367 496064
 chr12  chr13  chr14  chr15  chr16  chr17  chr18  chr19  chr20
558759 427010 365742 331501 354239 316396 322866 268235 323041
  chrX   chrY
391414   6539
> alphabetFrequency(Hsapiens$chr1)
       A        C        G        T        M        R        W
65570891 47024412 47016562 65668756        0        0        0
       Y        K        V        H        D        B        N
       0        0        0        0        0        0        0
       +
       0
> alphabetFrequency(HsWithSNPs$chr1)
       A        C        G        T        M        R        W
65306157 46833464 46825359 65403357    40477   150327    40710
       Y        K        V        H        D        B        N
  150117    41304   102527   125770   126323   102322      410
       +
       0
>
```

Acknowledgments

- Lars Steinmetz and his lab, especially:
  - Jonathan Landry
  - Julien Gagneur
  - Xing Xiaobin
- Eileen Furlong and her lab, especially:
  - Charles Girardot
- Wolfang Huber and his lab, especially:
  - Simon Anders
- Vladimir Benes and his Gene Core facility:
  - Tobias Rausch
  - Jonathon Blake