Analysis of SNP/Var-Seq Data with R/Bioconductor

Ruobai SUN

December 9, 2012

Variant (SNP/INDEL) Calling and Analysis Tools

Variant Call Format(VCF)

SNP/Var-Seq Analysis in R/Bioconductor

Variant Visualization in IGV

Outline

Variant (SNP/INDEL) Calling Tools

- Reads alignment: BOWTIE, BWA -> SAM/BAM format
- SAM/BAM Tools
  - Samtools Link Convert SAM <-> BAM ; Sort, index BAM file; Merge multiple BAM files
  - Picard Link SamtoFastq,FixMateInformation and more
  - Bio-SamTool Link
- Variant Calling
  - Samtools mpileup: input one or multiple BAM files, output SNPs and short INDELS in VCF file
  - GATK (Genome Analysis Toolkit) Link
- Variant Annotation
  - R Bioconductor package: VariantAnnotation package Link
  - snpEff tool Link
- Visualization
  - IGV Link
  - Galaxy Link

Outline

Variant Call Format (VCF)

- VCF Link is a standardised format
  - SNPs, insertions, deletions and structural variants
  - With rich annotations
  - Variants information across multiple samples
- Flexible format
  - Arbitrary tags can be introduced to describe new types of variants
  - User extensible annotation field supported

Variant Call Format (VCF)

- VCF consists of a header section and a data section.
- The header shows the name of each field and description.
- The data section is TAB delimited with each line consisting of fields
  - CHROM: Chromosome name
  - POS: 1-based position. For an indel, this is the position preceding the indel.
  - ID: Variant identifier.Usually the dbSNP rsID.
  - REF: Reference sequence at POS involved in the variant. For a SNP, it is a single base.
  - ALT: Comma delimited list of alternative seuqence(s).
  - QUAL: Phred-scaled probability of all samples being homozygous reference.
  - FILTER: Semicolon delimited list of filters that the variant fails to pass.
  - INFO: Semicolon delimited list of variant information.
  - FORMAT: Colon delimited list of the format of individual genotypes in the following fields.
  - Sample(s):Individual genotype information defined by FORMAT.

VCF Format Example

- Use a txt editor to open "data/var.raw.vcf"
- Samtools VCF format Link

```
##fileformat=VCFv4.1
##samtoolsVersion=0.1.18 (r982:295)
## This word was not found in the spelling dictionary. 'Raw read depth">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="# high-quality ref-forward bases, ref-reverse, alt-forward and alt-reverse bases">
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Root-mean-square mapping quality of covering reads">
##INFO=<ID=FQ,Number=1,Type=Float,Description="Phred probability of all samples being the same">
##INFO=<ID=AF1,Number=1,Type=Float,Description="Max-likelihood estimate of the first ALT allele frequency (assuming HWE)">
##INFO=<ID=AC1,Number=1,Type=Float,Description="Max-likelihood estimate of the first ALT allele count (no HWE assumption)">
##INFO=<ID=G3,Number=3,Type=Float,Description="ML estimate of genotype frequencies">
##INFO=<ID=HWE,Number=1,Type=Float,Description="Chi^2 based HWE test P-value based on G3">
##INFO=<ID=CLR,Number=1,Type=Integer,Description="Log ratio of genotype likelihoods with and without the constraint">
##INFO=<ID=UGT,Number=1,Type=String,Description="The most probable unconstrained genotype configuration in the trio">
##INFO=<ID=CGT,Number=1,Type=String,Description="The most probable constrained genotype configuration in the trio">
##INFO=<ID=PV4,Number=4,Type=Float,Description="P-values for strand bias, baseQ bias, mapQ bias and tail distance bias">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=PC2,Number=2,Type=Integer,Description="Phred probability of the nonRef allele frequency in group1 samples being larger (,smaller) than in group2.">
##INFO=<ID=PCHI2,Number=1,Type=Float,Description="Posterior weighted chi^2 P-value for testing the association between group1 and group2 samples.">
##INFO=<ID=QCHI2,Number=1,Type=Integer,Description="Phred scaled PCHI2.">
##INFO=<ID=PR,Number=1,Type=Integer,Description="# permutations yielding a smaller PCHI2.">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GL,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA genotypes (R=ref,A=alt)">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="# high-quality bases">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-value">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  SNPresults/SRR038850_aln.sorted.bam
Chr1    6324    .       TAAAA   TAAAAA  44.7    .       INDEL;DP=5;VDB=0.0743;AF1=1;AC1=2;DP4=0,1,2,2;MQ=35;FQ=-42.5;PV4=1,1,0.34,1     GT:PL:GQ        1/1:84,8,0:14
Chr1    20397   .       T       A       4.77    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=37;FQ=-30       GT:PL:GQ        0/1:33,3,0:3
Chr1    22126   .       A       T       16.9    .       DP=2;VDB=0.0278;AF1=1;AC1=2;DP4=0,0,2,0;MQ=37;FQ=-33    GT:PL:GQ        1/1:48,6,0:10
Chr1    26541   .       A       C       4.77    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=37;FQ=-30       GT:PL:GQ        0/1:33,3,0:3
Chr1    27540   .       G       T       28.8    .       DP=2;VDB=0.0278;AF1=1;AC1=2;DP4=0,0,2,0;MQ=37;FQ=-33    GT:PL:GQ        1/1:60,6,0:10
Chr1    37388   .       G       T       65.5    .       DP=4;VDB=0.0004;AF1=1;AC1=2;DP4=0,0,4,0;MQ=37;FQ=-39    GT:PL:GQ        1/1:98,12,0:21
Chr1    38554   .       G       T       4.13    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=37;FQ=-30       GT:PL:GQ        0/1:32,3,0:3
Chr1    46368   .       A       C       4.13    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=37;FQ=-30       GT:PL:GQ        0/1:32,3,0:3
Chr1    50704   .       A       T       4.77    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=37;FQ=-30       GT:PL:GQ        0/1:33,3,0:3
Chr1    55076   .       C       A       4.77    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=37;FQ=-30       GT:PL:GQ        0/1:33,3,0:3
Chr1    55782   .       A       G       15.2    .       DP=4;VDB=0.0306;AF1=0.5032;AC1=1;DP4=1,0,3,0;MQ=34;FQ=-8.64;PV4=1,0.061,0.33,0.11       GT:PL:GQ        0/1:45,0,19:22
```

Outline

Variant Calling and Analysis Exercises

- Read mapping with variant aware aligners
- SNP/INDEL calling
- Handling of standard variant data formats such as VCF
- Annotating variants with genomic context information
- Identification of synonymous/non-synonymous SNPs
- Injecting identified variants into reference genome/GFF

Data Sets

- To make the following sample code work, user can download the sample data Link into the directory of their current R session.
- It contains SRR038850 FASTQ file from the Sequence Read Archive (SRA) experiment published by Kaufman et al (2010, GSE20176 Link) ,the corresponding reference genome from Arabidopsis thaliana, the genome annotation file, and a vcf file which geneated with BWA and samtools mpileup linux command in Biocluster.

SNP/INDEL Calling with Linux Command

- For alignment and variants calling, the following commands could be ran in Biocluster (Linux) server.

Execute BWA and samtools mpileup with Linux commands

```
$ module load bwa
$ module load samtools
$ mkdir SNPresults
$ bwa index -a bwtsw data/tair10chr.fasta
$ bwa aln -t 5 data/tair10chr.fasta data/SRR038850.fastq SNPresults/SRR038850_aln_sa.sai
$ bwa samse data/tair10chr.fasta SNPresults/SRR038850_aln_sa.sai data/SRR038850.fastq
  > SNPresults/SRR038850_aln.sam
$ samtools view -bS SNPresults/SRR038850_aln.sam > SNPresults/SRR038850_aln.bam
$ samtools sorted SNPresults/SRR038850_aln.bam SNPresults/SRR038850_aln.sorted
$ samtools index SNPresults/SRR038850_aln.sorted.bam
$ samtools mpileup -ug -BQ0 -d10000000 -f data/tair10chr.fasta SNPresults/SRR038850_aln.
  |bcftools view -bvcg -> SNPresults/var.raw.bcf
$ bcftools view SNPresults/var.raw.bcf > SNPresults/var.raw.vcf
$ bgzip -c SNPresults/var.raw.vcf > SNPresults/var.raw.vcf.gz
$ tabix -p vcf SNPresults/var.raw.vcf.gz
```

Align Reads and Output Indexed Bam Files in R

```
dir.create("SNPresults")
targets <- read.csv("data/target.txt")
targets

Sample          Fastqfile        Type             Genome           Annotation      Aligner
SRR038850       SRR038850.fastq  singleEnd        tair10chr.fasta  Arabi.sqlite    bwa

 AlginmentParameters     variantscallParameters
 -t 5                    -BQ0 -d10000000

source("data/alignment.R")
source("data/variantscalling.R")
datadir <<-"data"
outputdir <- "SNPresults"
alignment(targets,outputdir)
samtoolsVariantCalling(targets,outputdir)
```

Read VCF file

Syntax

```
> library(VariantAnnotation)
> library(GenomicFeatures)
> ## Pleas make sure exact data.zip under your current work directory
> vcf <- readVcf("data/var.raw.vcf.gz","TAIR10")
> vcf
```

```
class: VCF
dim: 413 1
genome: TAIR10
exptData(1): header
fixed(4): REF ALT QUAL FILTER
info(18): DP DP4 ... PR VDB
geno(6): GT GQ ... SP PL
rownames(413): Chr1:6324 Chr1:20397 ... Chr5:17353752 Chr5:24404293
rowData values names(1): paramRangeID
colnames(1): SNPresults/SRR038850_aln.sorted.bam
colData names(1): Samples
```

Read Header in VCF

Syntax

```
> #Extract the header information stored in the exptData slot
> hdr <-exptData(vcf)[["header"]]
> hdr

class: VCFHeader
samples(1): SNPresults/SRR038850_aln.sorted.bam
meta(2): fileformat samtoolsVersion
fixed(0):
info(18): DP DP4 ... PR VDB
geno(6): GT GQ ... SP PL

> #Explore it with the fixed, info and geno accessors
> head(fixed(hdr),1)

SimpleDataFrameList of length 0
names(0):

> head(info(hdr),2)

DataFrame with 2 rows and 3 columns
         Number        Type
    <character> <character>
DP            1     Integer
DP4           4     Integer
                                                                  Description
                                                                  <character>
DP                                                              Raw read depth
DP4 # high-quality ref-forward bases, ref-reverse, alt-forward and alt-reverse bases

> head(geno(hdr),2)

DataFrame with 2 rows and 4 columns
        Number        Type      Description           A
   <character> <character>      <character> <character>
GT           1      String         Genotype          NA
GQ           1     Integer Genotype Quality          NA
```

Read Rowdata of VCF

Syntax

```
> head(rowData(vcf))
```

```
GRanges with 6 ranges and 1 metadata column:
             seqnames          ranges strand | paramRangeID
                <Rle>       <IRanges>  <Rle> |     <factor>
   Chr1:6324     Chr1 [ 6324,  6328]      * |          <NA>
  Chr1:20397     Chr1 [20397, 20397]      * |          <NA>
  Chr1:22126     Chr1 [22126, 22126]      * |          <NA>
  Chr1:26541     Chr1 [26541, 26541]      * |          <NA>
  Chr1:27540     Chr1 [27540, 27540]      * |          <NA>
  Chr1:37388     Chr1 [37388, 37388]      * |          <NA>
  ---
  seqlengths:
   Chr1 Chr2 Chr3 Chr4 Chr5
     NA   NA   NA   NA   NA
```

Read Fixed Fields

- The REF, ALT, QUAL and FILTER fields can be accessed together with fixed accessor
- Or individually with ref, alt, qual and filt accessors.

Syntax

```
> head(fixed(vcf),3)
```

```
GRanges with 3 ranges and 5 metadata columns:
             seqnames          ranges strand | paramRangeID              REF
                <Rle>       <IRanges>  <Rle> |     <factor> <DNAStringSet>
  Chr1:6324      Chr1 [ 6324,  6328]      * |         <NA>           TAAAA
 Chr1:20397      Chr1 [20397, 20397]      * |         <NA>               T
 Chr1:22126      Chr1 [22126, 22126]      * |         <NA>               A
                          ALT      QUAL      FILTER
             <DNAStringSetList> <numeric> <character>
  Chr1:6324           ########     44.70           .
 Chr1:20397           ########      4.77           .
 Chr1:22126           ########     16.90           .
 ---
 seqlengths:
  Chr1 Chr2 Chr3 Chr4 Chr5
    NA   NA   NA   NA   NA
```

Read ALT Field

Syntax

```
> alternate <- alt(vcf)
> alternate

DNAStringSetList of length 413
[[1]] TAAAAA
[[2]] A
[[3]] T
[[4]] C
[[5]] T
[[6]] T
[[7]] T
[[8]] C
[[9]] T
[[10]] A
...
<403 more elements>
```

Read ALT Field

Syntax

```
> ## number of ALT values per variant
> unique(elementLengths(alternate))

[1] 1 2

> head(unlist(alternate))

  A DNAStringSet instance of length 6
    width seq
[1]     6 TAAAAA
[2]     1 A
[3]     1 T
[4]     1 C
[5]     1 T
[6]     1 T
```

Read GENO Field

Syntax

```
> geno(vcf)

SimpleList of length 6
names(6): GT GQ GL DP SP PL

> geno(vcf)$DP[1:3,]

 Chr1:6324 Chr1:20397 Chr1:22126
        NA          NA          NA
```

Variants Filtering

Syntax

```
> index1 <-lapply(as.data.frame(info(vcf),row.names=NULL)[["DP4"]],
+        function(x) (x[3]+x[4])>(x[1]+x[2]) & (x[3]+x[4])>3)
> index2 <- as.data.frame(info(vcf),row.names=NULL)[["MQ"]]>10
> index <- which(index1==TRUE & index2==TRUE)
> filtered_fixed = fixed(vcf)[index,]
> length(filtered_fixed)

[1] 31

> length(fixed(vcf))

[1] 413
```

Variant Annotation

Syntax

```
> #ArabiTranDB <- makeTranscriptDbFromGFF(file="TAIR10.gff3",
> #                       format="gff3",
> #                       dataSource="TAIR 10",
> #                       species="Arabidopsis")
>
> #saveDb(ArabiTranDB,file="Arabi.sqlite")
> txdb <- loadDb("data/Arabi.sqlite")
> txdb## confirm seqlevels are the same

TranscriptDb object:
| Db type: TranscriptDb
| Supporting package: GenomicFeatures
| Data source: TAIR 10
| Genus and Species: Arabodpsis
| miRBase build ID: NA
| transcript_nrow: 143
| exon_nrow: 636
| cds_nrow: 601
| Db created by: GenomicFeatures package from Bioconductor
| Creation time: 2012-12-05 10:51:47 -0800 (Wed, 05 Dec 2012)
| GenomicFeatures version at creation time: 1.10.0
| RSQLite version at creation time: 0.11.2
| DBSCHEMAVERSION: 1.0
```

Variant Annotation

Syntax

```
> ## confirm seqlevels are the same
> intersect(seqlevels(vcf), seqlevels(txdb))

[1] "Chr1" "Chr2" "Chr3" "Chr4" "Chr5"

> #vcf <- renameSeqlevels(vcf, c("chr1"="Chr1"))
>
> rd <- rowData(vcf)[index,]
> loc <- locateVariants(rd, txdb, CodingVariants())
> loc[1:2,]

GRanges with 2 ranges and 5 metadata columns:
      seqnames          ranges strand | LOCATION   QUERYID      TXID     CDSID
         <Rle>       <IRanges>  <Rle> | <factor> <integer> <integer> <integer>
  [1]     Chr2 [ 1906,  1909]      * |   coding         4        26       307
  [2]     Chr3 [14338, 14338]      * |   coding        13        35       365
           GENEID
      <character>
  [1]   AT2G01008
  [2]   AT3G01050
  ---
  seqlengths:
   Chr1 Chr2 Chr3 Chr4 Chr5
     NA   NA   NA   NA   NA
```

Variant Annotation

Syntax

```
> allvar <- locateVariants(rd, txdb, AllVariants())
> allvar[1:3,]

GRanges with 3 ranges and 7 metadata columns:
   seqnames          ranges strand | LOCATION   QUERYID      TXID     CDSID
      <Rle>       <IRanges>  <Rle> | <factor> <integer> <integer> <integer>
       Chr1 [ 6324,  6328]      * |   intron         1         2      <NA>
       Chr1 [37388, 37388]      * |  fiveUTR         2         8      <NA>
       Chr1 [37388, 37388]      * | threeUTR         2         8      <NA>
        GENEID   PRECEDEID    FOLLOWID
   <character> <character> <character>
     AT1G01020        <NA>        <NA>
     AT1G01060        <NA>        <NA>
     AT1G01060        <NA>        <NA>
  ---
  seqlengths:
   Chr1 Chr2 Chr3 Chr4 Chr5
     NA   NA   NA   NA   NA
```

```
> dir.create("SNPresults")
> alternate <- as.character(unlist(alt(vcf)))[values(allvar)[["QUERYID"]]]
> reference <- as.character(ref(vcf))[values(allvar)[["QUERYID"]]]
> snpquality <- values(fixed(vcf))[["QUAL"]][values(allvar)[["QUERYID"]]]
> DP4 <-do.call(rbind,as.data.frame(info(vcf),row.names=NULL)[["DP4"]])[values(allvar)[["QUERYID"]],]
> DP<- as.data.frame(info(vcf),row.names=NULL)[["DP"]][values(allvar)[["QUERYID"]]]
> MQ <-as.data.frame(info(vcf),row.names=NULL)[["MQ"]][values(allvar)[["QUERYID"]]]
> annotable <- cbind(as.data.frame(allvar,row.names = NULL), reference,alternate,snpquality,DP,DP4,MQ)[-c(5
> write.table(annotable,"SNPresults/var_annotation.xls",row.names=FALSE,
+    col.names=c("Chr","Start","End","With","Location","GeneID","PrecedID","FollowID","Reference","Alternat
```

Variant Annotation

Syntax

```
> ## Did any variants match more than one gene
> table(sapply(split(values(allvar)[["GENEID"]], values(allvar)[["QUERYID"]]),
+       function(x) length(unique(x)) > 1))

FALSE  TRUE
   26     5
```

Variant Annotation

Syntax

```
> ## Summarize the number of variants by gene
> idx <- sapply(split(values(allvar)[["QUERYID"]], values(allvar)[["GENEID"]]),
+     unique)
> sapply(idx, length)[1:5]
```

```
AT1G01020 AT1G01060 AT2G01008 AT2G01023 AT3G01050
        1         1         1         1         1
```

Variant Annotation

Syntax

```
>  ## Summarize variant location by gene
> sapply(names(idx),
+        function(nm) {
+            d <- allvar[values(allvar)[["GENEID"]] %in% nm, c("QUERYID", "LOCATI
+            table(values(d)[["LOCATION"]][duplicated(d) == FALSE])
+        })[,1:5]
```

	AT1G01020	AT1G01060	AT2G01008	AT2G01023	AT3G01050
spliceSite	0	0	0	0	0
intron	1	0	0	0	0
fiveUTR	0	1	0	0	0
threeUTR	0	0	0	0	0
coding	0	0	1	0	1
intergenic	0	0	0	0	0
promoter	0	0	0	1	0

Predict Coding Variant

Syntax

```
> library(BSgenome)
> #source("http://bioconductor.org/biocLite.R")
> #biocLite("BSgenome.Athaliana.TAIR.TAIR9")
> library(BSgenome.Athaliana.TAIR.TAIR9)


> coding <- predictCoding(vcf, txdb, Athaliana)
> coding <- coding[unique(names(rd))%in% names(coding)]
> write.table(as.data.frame(coding,row.names=NULL)[,-c(5,6,12,13,14)],"SNPresult
```

Predict Coding Variant

Syntax

```
> nms <- names(coding)
> idx <- values(coding)[["CONSEQUENCE"]] == "nonsynonymous"
> nonsyn <- coding[idx]
> head(nonsyn,2)

GRanges with 2 ranges and 13 metadata columns:
             seqnames         ranges strand | paramRangeID      varAllele
                <Rle>      <IRanges>  <Rle> |     <factor> <DNAStringSet>
  Chr3:11475     Chr3 [11475, 11475]      + |         <NA>              C
  Chr4:48772     Chr4 [48772, 48772]      + |         <NA>              A
                  CDSLOC              PROTEINLOC   QUERYID        TXID
                <IRanges> <CompressedIntegerList> <integer> <character>
  Chr3:11475 [1188, 1188]                    396       117          34
  Chr4:48772 [ 289,  289]                     97       255          70
                  CDSID      GENEID   CONSEQUENCE       REFCODON       VARCODON
             <integer> <character>      <factor> <DNAStringSet> <DNAStringSet>
  Chr3:11475       359   AT3G01040 nonsynonymous            TTT            TTA
  Chr4:48772       831   AT4G00130 nonsynonymous            GCT            TCT
                    REFAA         VARAA
             <AAStringSet> <AAStringSet>
  Chr3:11475             F             L
  Chr4:48772             A             S
  ---
  seqlengths:
   Chr1 Chr2 Chr3 Chr4 Chr5
     NA   NA   NA   NA   NA
```

Inject Variants to Update Genome/GFF

Syntax

```
> source("data/genomeUpdate_Fct.R")
> chr1 <- DNAString(paste(sample(c("A", "T", "G", "C"), 200, replace=T), collapse=""))
> injDF <- data.frame(Chr=c(1, 1, 1, 1, 1, 1), Start=c(1, 10, 20, 30, 40, 50), End=c(1, 10, 25, 30, 40, 60)
> injDF[1:3,]

  Chr Start End Type Value
1   1     1   1  SNP     A
2   1    10  10  Ins TTTTT
3   1    20  25  Del

> features <- data.frame(Start=c(1, 20, 90, 150, 180), End=c(40, 30, 105, 160, 200), Feature="mRNA")
> modchr <- modChr(chr=chr1, injDF=injDF)
> modchr$indexmap[9:17]

 9 10 11 12 13 14 15 16 17
 9 10 10 10 10 10 10 11 12

> mRNAnew <- featureExtr(indexnew=modchr$indexmap, modchr=modchr$chrmod, features=features)
> modFeatures(injDF=injDF, features=features, modchr=modchr, type="coor")

  start end Feature
6     1  45    mRNA
2    25  29    mRNA
3    84  99    mRNA
4   144 154    mRNA
5   174 194    mRNA
```

Subsetting VCF with Fields

Syntax

```
> ## Return "ALT" from 'fixed',"DP4" from "info",  and "PL" from 'geno'
> svp <- ScanVcfParam(fixed="ALT",info="DP4", geno="GT")
> vcf_flds <- readVcf("data/var.raw.vcf.gz", "TAIR10", svp)
> geno(vcf_flds)

SimpleList of length 1
names(1): GT

> head(fixed(vcf_flds),3)

GRanges with 3 ranges and 3 metadata columns:
             seqnames         ranges strand | paramRangeID            REF
                <Rle>      <IRanges>  <Rle> |     <factor> <DNAStringSet>
   Chr1:6324     Chr1 [ 6324,  6328]      * |         <NA>          TAAAA
  Chr1:20397     Chr1 [20397, 20397]      * |         <NA>              T
  Chr1:22126     Chr1 [22126, 22126]      * |         <NA>              A
                           ALT
             <DNAStringSetList>
   Chr1:6324           ########
  Chr1:20397           ########
  Chr1:22126           ########
  ---
  seqlengths:
   Chr1 Chr2 Chr3 Chr4 Chr5
     NA   NA   NA   NA   NA
```

Subsetting VCF with Ranges

Syntax

```
> which <- RangesList(Chr1=IRanges(1, 100000),
+                     Chr2=IRanges(100, 100000),
+                     Chr3=IRanges(100, 100000),
+                     Chr4=IRanges(1, 100000),
+                     Chr5=IRanges(1, 100000))
> vcf <- readVcf("data/var.raw.vcf.gz","TAIR10",ScanVcfParam(which=which))
```

Inject Variants to Update Genome/GFF

Syntax

```
> genome<-readDNAStringSet("data/tair10chr_trunc.fasta")
> index1 <-lapply(as.data.frame(info(vcf),row.names=NULL)[["DP4"]],
+        function(x) (x[3]+x[4])>(x[1]+x[2]) & (x[3]+x[4])>3)
> index2 <- as.data.frame(info(vcf),row.names=NULL)[["MQ"]]>10
> index <- which(index1==TRUE & index2==TRUE)
> injDF <-convertinjDF(vcf)
> injDF <-injDF[index,]
> i="Chr1"
> modchr <- modChr(genome[which(names(genome)==i)][[1]],injDF[which(injDF[,1]==i),])
> injDF[1,]

   chr Start  End Type  Value
1 Chr1  6324 6324  Ins TAAAAA

> modchr$indexmap[6324:6337]# insert example

6324 6325 6326 6327 6328 6329 6330 6331 6332 6333 6334 6335 6336 6337
6324 6324 6324 6324 6324 6324 6324 6325 6326 6327 6328 6329 6330 6331

> chrinj <- DNAStringSet()
> for (i in names(table(injDF[,1]))) {
+ modchr <- modChr(genome[which(names(genome)==i)][[1]],injDF[which(injDF[,1]==i),])
+ chrinj <- c(chrinj, DNAStringSet(modchr$chrmod))
+ }
> ## write to a new genome
> names(chrinj) <-names(table(injDF[,1]))
> writeXStringSet(chrinj, "SNPresults/injected_tair10chr_trunc.fa")
```

Session Information

```
> sessionInfo()

R version 2.15.1 (2012-06-22)
Platform: x86_64-unknown-linux-gnu (64-bit)
locale:
[1] C

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

other attached packages:
 [1] BSgenome.Athaliana.TAIR.TAIR9_1.3.18 BSgenome_1.26.1
 [3] GenomicFeatures_1.10.0               AnnotationDbi_1.20.1
 [5] Biobase_2.18.0                       VariantAnnotation_1.4.1
 [7] Rsamtools_1.10.1                     Biostrings_2.26.2
 [9] GenomicRanges_1.10.2                 IRanges_1.16.3
[11] BiocGenerics_0.4.0

loaded via a namespace (and not attached):
 [1] DBI_0.2-5          RCurl_1.95-1.1     RSQLite_0.11.2     XML_3.95-0.1
 [5] biomaRt_2.14.0     bitops_1.0-4.1     parallel_2.15.1    rtracklayer_1.18.0
 [9] stats4_2.15.1      tools_2.15.1       zlibbioc_1.4.0
```

Outline

Variant Visualization in IGV

- Load TAIR10 genome to IGV
- Load BAM file SRR038850_aln.sorted.bam to IGV: Load from URL: `http://biocluster.ucr.edu/~rsun/workshop/SNPINDEL/SRR038850_aln.sorted.bam`
- Index vcf file with igvtool in IGV : Select "Run igv tool" in Tool bar of IGV, "Command" is "Index", "Input" is "data/var.raw.vcf"
- Load vcf file to IGV: Load from file: data/var.raw.vcf

Zoom In Screen

Go to: Chr2:3,438,323-3,438,363