BEDtools, software installation

Bioinformatics Applications (PLPTH813)

Sanzhen Liu

4/15/2021

Outline

- BED format
- BEDtools and examples
- Software installation

BEDtools (I)

intersect	**Find overlapping intervals in various ways.**
window	Find overlapping intervals within a window
closest	**Find the closest, potentially non-overlapping interval.**
coverage	Compute the coverage over defined intervals.
map	Apply a function to a column
genomecov	Compute the coverage over an entire genome.
merge	Combine overlapping/nearby intervals
cluster	Cluster (but don't merge) overlapping/nearby intervals.
complement	Extract intervals _not_ represented by an interval file.
shift	Adjust the position of intervals.
subtract	Remove intervals based on overlaps b/w two files.
slop	Adjust the size of intervals.
flank	**Create new intervals from flanks of existing intervals.**
sort	Order the intervals in a file.
random	Generate random intervals in a genome.
shuffle	Randomly redistribute intervals in a genome.
sample	Sample random records from file using reservoir sampling.
spacing	Report the gap lengths between intervals in a file.
annotate	Annotate coverage of features from multiple files.

BEDtools (II) - Fasta manipulation

getfasta Use intervals to extract sequences from a FASTA file.

maskfasta Use intervals to mask sequences from a FASTA file.

nuc Profile the nucleotide content of intervals

Beocat: module load BEDTools

BED format (Tab-separated file) (I)

The first three required BED fields are:

1. **chrom** - the chromosome
2. **chromStart** - the starting position; 0-based
3. **chromEnd** - the ending position; 1-based

e.g., the first 100 bases of chromosome 1

```
chr1 0  100
```

BED format (II)

The additional optional BED fields are:

4. name - Defines the name of the BED line.

5. score - A score between 0 and 1000

6. strand - Defines the strand - either '+' or '-'

…

e.g., the first 100 bases of chromosome 1

```
chr1 0  100  region1 .  +
chr1 100  200  region2 .  -
```

…

BED format (III)

- Other optional fields

(could be flexible and more fields)

thickStart - coordinate to start drawing a solid rectangle

thickEnd - coordinate to stop drawing a solid rectangle

itemRgb - an RGB colour value (e.g. 0,0,255).

e.g.,

```
chr1    0       100     region1 .       +       0       100     255,0.0
```

Extract promoter sequences of genes (I)

Required input information

1. a BED file of genes (.bed)
2. Genome sequences (.fasta)
3. Chromosome/contig lengths (.length)
4. length of promoters to be extracted

How To Use Bedtools To Extract Promoters?

Extract promoter sequences of genes (II)

```
bed=genes.bed
ref=ref.fasta
clen=chrs.length
promoter_len=2000
out=genes.promoter

# generate a BED file
bedtools flank -i $bed -g $clen \
  –l $promoter_len -r 0 -s > $out.bed

# extract sequence
bedtools getfasta -s -fi $ref \
  -bed $out.bed -fo ${out}.fasta
```

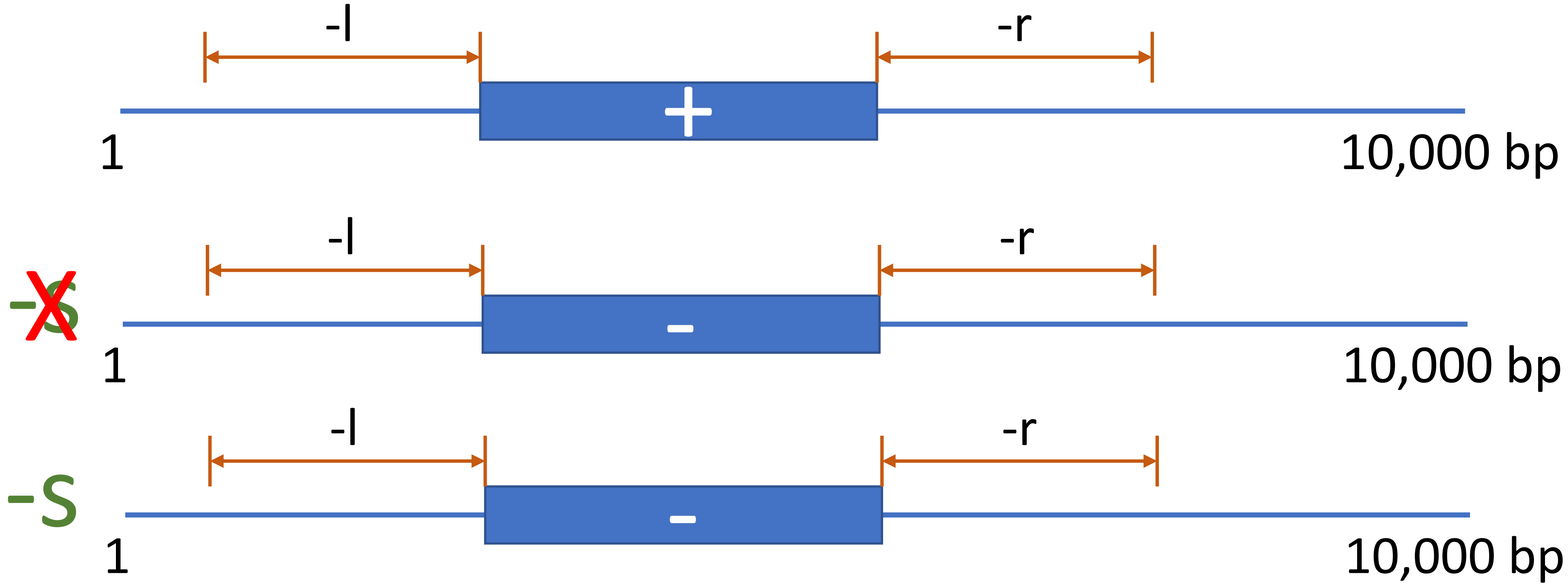
BEDtools - flank

-b bp flanking in each direction.

-l bp flanking from start coordinate.

-r bp flanking from end coordinate

-s define "start" and "end" based on strand.



Promoter - example

1. ref.fas

```
>ref
AAAAAAAAAAAAAAAAAAAACCCCCCCCCCCCCCCCCCCCAAGGGGGGGGGGGGGGGGGG
```

2. gene.bed

```
ref   20   40   a1   0.5   +
ref   20   40   a2   0.5   -
```

3. ref.length

```
ref   60
```

```
bedtools flank -i gene.bed -g ref.length \
  -l 20 -r 0 -s > gene.promoter.bed
```

Output:

```
  ref   0  20 a1 0.5  +
  ref   40 60 a2 0.5  -
```

BEDtools - getfasta

```
-fi         Input FASTA file
-fo         Output file
-bed        BED/GFF/VCF file of ranges to extract from -fi
-s          Force strandedness. If the strand is minus(-), the
sequence will be reverse complemented.
```

```
bedtools getfasta -s -fi ref.fas \
-bed gene.promoter.bed \
-fo gene.promoter.fas
```

1. ref.fas

```
>ref
AAAAAAAAAAAAAAAAAAAACCCCCCCCCCCCCCCCCCCCAAGGGGGGGGGGGGGGGGGG
```

2. gene.promoter.bed

```
ref   0    20   a1   0.5   +
ref   40   60   a2   0.5   -
```

Output:

```
>ref:0-20(+)
AAAAAAAAAAAAAAAAAAAA
>ref:40-60(-)
CCCCCCCCCCCCCCCCCCTT
```

intersect (I)

cat d1.bed

```
chr1   10    100   a1   .   +
chr1   200   300   a2   .   +
chr1   500   550   a3   .   +
```

cat d2.bed

```
chr1   10    20    a1   .   +
chr1   150   250   a2   .   +
chr1   600   750   a3   .   +
```

intersect (II)

cat d1.bed

```
chr1	10	100	a1	.	+
chr1	200	300	a2	.	+
chr1	500	550	a3	.	+
```

cat d2.bed

```
chr1	10	20	a1	.	+
chr1	150	250	a2	.	+
chr1	600	750	a3	.	+
```

```
bedtools intersect -a d1.bed -b d2.bed
chr1 10 20 a1 . +
chr1 200 250 a2 . +
```

intersect (III)

cat d1.bed

```
chr1	10	100	a1	.	+
chr1	200	300	a2	.	+
chr1	500	550	a3	.	+
```

cat d2.bed

```
chr1	10	20	a1	.	+
chr1	150	250	a2	.	+
chr1	600	750	a3	.	+
```

```
bedtools intersect -a d1.bed -b d2.bed -wo
```

```
chr1	10	100	a1	.	+	chr1	10	20	a1
	.	+	10
chr1	200	300	a2	.	+	chr1	150	250	a2
	.	+	50
```

-wo: write the original A and B entries plus the number of base pairs of overlap between the two features.

intersect (IV)

cat d1.bed

```
chr1	10	100	a1	.	+
chr1	200	300	a2	.	+
chr1	500	550	a3	.	+
```

cat d2.bed

```
chr1	10	20	a1	.	+
chr1	150	250	a2	.	+
chr1	600	750	a3	.	+
```

```
bedtools intersect -a d1.bed -b d2.bed -wao
chr1	10	100	a1	.	+	chr1	10	20	a1
	.	+	10
chr1	200	300	a2	.	+	chr1	150	250	a2
	.	+	50
chr1	500	550	a3	.	+	.	-1	-1	.
	-1	.	0
```

-wao: write the original A and B entries plus the number of base pairs of overlap between the two features. A features w/o overlap are also reported.

coverage

```
bedtools coverage -abam $bam -b $bed
```

```
Bed input: Interval_1   1  1645210 -3.84
           …
Output:
```

```
                                                1        2        3        4
Interval_1    1    16452          10   -3.84    5432     16302    16451    0.9909
```

1. Read number
2. Coverage (bp)
3. Original length
5. Coverage (%)

closest

```
# find the closest, non-overlapping gene for each peak interval

bedtools closest -a peak.bed -b genes.bed -io > peak.near.genes.bed
```

slop & complement

```
# Add 500 bp up and downstream of each probe
bedtools slop -i probes.bed -b 500 > p.500bp.bed

# Get a BED file of all regions not covered by the probes (+500 bp up/down)
bedtools complement -i p.500bp.bed -g hg18.genome > p.500bp.complement.bed
```

window

#Report genes within 10kb upstream or downstream of CNVs.

```
bedtools window -a CNVs.bed -b genes.bed -w 10000
```

# Report genes within 10kb upstream or 5kb downstream of CNVs.

```
bedtools window -a CNVs.bed -b genes.bed -l 10000 -r 5000
```

#Report SNPs within 5kb upstream or 1kb downstream of genes. Define upstream and downstream based on strand.

```
bedtools window -a genes.bed -b snps.bed -l 5000 -r 1000 -sw
```

merge

```
# Merge overlapping repetitive elements into a single entry.
bedtools merge -i repeatMasker.bed
```

```
# Merge overlapping repetitive elements into a single entry, returning the number of entries merged.
bedtools merge -i repeatMasker.bed -n
```

```
# Merge nearby (within 1kb) repetitive elements into a single entry.
bedtools merge -i repeatMasker.bed -d 1000
```

Outline

- BED format
- BEDtools and examples
- Software installation

HiSat2 (example: compiled package)

```
wget https://cloud.biohpc.swmed.edu/index.php/s/hisat2-220-Linux_x86_64/download

mv download hisat2-220-Linux_x86_64.zip
unzip hisat2-220-Linux_x86_4.zip
cd hisat2-2.2.0/

# Edit ~/.bashrc
# PATH=$PATH:~/software/hisat2/hisat2-2.2.0:...
source ~/.bashrc
which hisat2
#~/software/hisat2/hisat2-2.2.0/hisat2
```

bwa (uncompiled package)

```
wget https://sourceforge.net/projects/bio-bwa/files/bwa-0.7.17.tar.bz2/download
mv download bwa-0.7.17.tar.bz2
tar -xf bwa-0.7.17.tar.bz2
cd bwa-0.7.17
# compile
make
# change ~/.bashrc
PATH=$PATH:~/software/bwa/bwa-0.7.17:~/software/hisat2/hisat2-2.2.0:…
source ~/.bashrc
bwa
```

conda

Conda is an open-source package management system and environment management system.

Conda quickly installs, runs and updates packages and their dependencies.

conda installation

```
# download conda software
wget https://repo.anaconda.com/archive/Anaconda3-2020.11-Linux-x86_64.sh

# installation
sh Anaconda3-2020.11-Linux-x86_64.sh

# with -u if a previous installation exists
# sh Anaconda3-2020.11-Linux-x86_64.sh -u
```

Software installation via conda

```
conda create -n test
conda activate <env_name>
conda install xxx
# install package through bioconda channel
conda install -c bioconda xxx
```

```
     conda-env list

     conda activate test
```