Week 10, Lecture 20

István Albert

Biochemistry and Molecular Biology
and Bioinformatics Consulting Center

Penn State

BEDOPS

A suite of tools to address common questions raised in genomic studies — mostly with regard to overlap and proximity relationships between data sets — BEDOPS aims to be scalable and flexible, facilitating the efficient and accurate analysis and management of large-scale genomic data.

downloads

Current download: v1.2.5 / October 13, 2012 (changes)

A companion set of documentation, conversion scripts and sample input files is also available for offline browsing.

It offers a complementary tools and approaches — very nice examples and tutorials that demonstrate the thought process that goes with interval analysis.
Back to BedTools –
most important tool: \texttt{intersect}

• Some tools are easier to rewrite: \texttt{slop}, \texttt{flank}, \texttt{complement} (although can be tricky)

• \texttt{Intersect} is the bread-and-butter,
  It would take substantial work to write a better intersect than what \texttt{bedtools} has
Tool: bedtools intersect (aka intersectBed)
Version: v2.16.2
Summary: Report overlaps between two feature files.

Usage: bedtools intersect [OPTIONS] -a <bed/gff/vcf> -b <bed/gff/vcf>

Options:

- **-abam**: The A input file is in BAM format. Output will be BAM as well.
- **-ubam**: Write uncompressed BAM output. Default writes compressed BAM.
- **-bed**: When using BAM input (-abam), write output as BED. The default is to write output in BAM when using -abam.
- **-wa**: Write the original entry in A for each overlap.
- **-wb**: Write the original entry in B for each overlap.
  - Useful for knowing what A overlaps. Restricted by -f and -r.
- **-loj**: Perform a "left outer join". That is, for each feature in A report each overlap with B. If no overlaps are found, report a NULL feature for B.
Using the genes feature file

```bash
ialbert@porthos ~/work/lec20
$ head -3 simple.gff
chrI  SGD  chromosome  1  230218 .  .  .  .
chrI  SGD  telomeric_repeat  1  62  .  -  .  .
chrI  SGD  telomere  1  801  .  -  .  .

ialbert@porthos ~/work/lec20
$ cat simple.gff | awk '$3=="gene" { print $0 }' > genes.gff

ialbert@porthos ~/work/lec20
$ head -3 genes.gff
chrI  SGD  gene  335  649  .  +  .  .
chrI  SGD  gene  538  792  .  +  .  .
chrI  SGD  gene  1807  2169  .  -  .  .
```
Running the intersection

-wo flag writes entry A then entry B followed by the overlap
Quite a few features – strand, overlap fraction and others

- **-c**
  For each entry in A, report the number of overlaps with B.
  - Reports 0 for A entries that have no overlap with B.
  - Overlaps restricted by -f and -r.

- **-v**
  Only report those entries in A that have _no overlaps_ with B.
  - Similar to "grep -v" (an homage).

- **-f**
  Minimum overlap required as a fraction of A.
  - Default is 1E-9 (i.e., 1bp).
  - FLOAT (e.g. 0.50)

- **-r**
  Require that the fraction overlap be reciprocal for A and B.
  - In other words, if -f is 0.90 and -r is used, this requires that B overlap 90% of A and A _also_ overlaps 90% of B.

- **-s**
  Require same strandedness. That is, only report hits in B that overlap A on the _same_ strand.
  - By default, overlaps are reported without respect to strand.

- **-S**
  Require different strandedness. That is, only report hits in B that overlap A on the _opposite_ strand.
  - By default, overlaps are reported without respect to strand.
Intersect variants: window, closest

- **bedtools window**

  Examines a "window" around each feature in A and reports all features in B that overlap the window.

- **bedtools closest**

  For each feature in A, finds the closest feature (upstream or downstream) in B.

  **Window**: extend or flank the original intervals and use intersect.
  **Closest**: intersect then sort by gene name then by overlap.
Other helpful bedtool commands: 

**getfasta** — extracts sequences

```
ialbert@porthos ~/work/lec20
$ ~/bin/bedtools getfasta

Tool: bedtools getfasta (aka fastaFromBed)
Version: v2.16.2
Summary: Extract DNA sequences into a fasta file based on feature coordinates.

Usage: bedtools getfasta [OPTIONS] -fi <fasta> -bed <bed/gff/vcf> -fo <fasta>

Options:
- `-fi` Input FASTA file
- `-bed` BED/GFF/VCF file of ranges to extract from `-fi`
- `-fo` Output file (can be FASTA or TAB-delimited)
- `-name` Use the name field for the FASTA header
- `-tab` Write output in TAB delimited format.
  - Default is FASTA format.

- `-s` Force strandedness. If the feature occupies the antisense, strand, the sequence will be reverse complemented.
  - By default, strand information is ignored.
```
Get the sequence to every feature annotated as **binding_site**

```bash
ialbert@porthos ~/work/lec20
$ cat simple.gff | awk ' $3=="binding_site" { print $0 } ' > binding.gff

ialbert@porthos ~/work/lec20
$ head -3 binding.gff
chrI  SGD  binding_site  532  544  .  .  .
chrI  SGD  binding_site  229670  229682  .  +  .  .
chrII SGD  binding_site  6339  6351  .  .  .

ialbert@porthos ~/work/lec20
$ ~/bin/bedtools getfasta -fi refs/sc.fa -bed binding.gff -fo binding.fa

ialbert@porthos ~/work/lec20
$ head -4 binding.fa
>chrI:531-544
ATCATATTGCACG
>chrI:229669-229682
CGTGTATGGTGAT

ialbert@porthos ~/work/lec20
$ note the coordinate system change!
```
bedtools nuc

ialbert@porthos ~/work/lec20
$ ~/bin/bedtools nuc

Tool: bedtools nuc (aka nucBed)
Version: v2.16.2
Summary: Profiles the nucleotide content of intervals in a fasta file.

Usage: bedtools nuc [OPTIONS] -fi <fasta> -bed <bed/gff/vcf>

Options:
- fi Input FASTA file
- bed BED/GFF/VCF file of ranges to extract from -fi
- s Profile the sequence according to strand.
- seq Print the extracted sequence
- pattern Report the number of times a user-defined sequence is observed (case-sensitive).
- C Ignore case when matching -pattern. By default, case matters.

Output format:
The following information will be reported after each BED entry:
1) %AT content
Very potent tools once we understand what it does. Groups on a column and applies an operator to a different column.
More on groupby

• data must be sorted on the grouping column

• Valid operators: sum, mean, median, mode, min, max (see help for more)

• Example: Find the average length of each feature in the features.gff file
Using `groupby`
Homework 20: refine homework 17

• Did you get fewer or more SNPs than expected?

• Current homework will be to match the called SNPs to the expectations
The full data script (see website).

1. Generates mutated data
2. Aligns mutated data
3. Creates the BAM file
4. Generates genotype information
5. Calls the SNPs

**Question**

Which SNPs match the expectation?
Which SNPs do not not
Transform the expected SNPs output

- wgsim mutation output does not produce a known interval format $\Rightarrow$ transform it to one.

- What should it be? Ideally a VCF but that may be too complicated for our needs

- Keep it simple $\Rightarrow$ we could also turn it into something simple like GFF
Generate a GFF file from the mutations

We are somewhat misusing the GFF format (for example the source/type are invalid)

(this above probably deeply disappoints Lincoln Stein and other people that designed the GFF)
Intersect the two files

```bash
ialbert@porthos ~/work/lec20
$ ~/bin/bedtools intersect -a mut.gff -b bwa.vcf -wo | head -3
chrI  C  A  2518  2518  .  .  .  .  chrI  2518  6
8.2  .  DP=11;VDB=0.0578;AF1=1;AC1=2;DP4=1,0,4,6;MQ=60;FQ=-43;PV4=0.45,0.38,1,1 G
T:PL:DP:GQ 1/1:101,16,0:11:29 1
chrI  G  R  3622  3622  .  +  .  .  chrI  3622  2
9  .  DP=10;VDB=0.0407;AF1=0.5005;AC1=1;DP4=0,4,3,3;MQ=58;FQ=3.02;PV4=0.2,1,0.2
2,1  GT:PL:DP:GQ 0/1:59,0,27:10:30 1
chrI  G  S  8811  8811  .  +  .  .  chrI  8811  3
7  .  DP=15;VDB=0.0460;AF1=0.5;AC1=1;DP4=4,3,4,4;MQ=55;FQ=29.6;PV4=1,1,0.32,0.1
1  GT:PL:DP:GQ 0/1:67,0,57:15:60 1

ialbert@porthos ~/work/lec20
$ ```
Refine homework 17. Use the VCF file and mutations file from that homework. Show the code and report the numbers for

- **TP (True Positives):** number of expected SNPs that you have found
- **FP (False Positives) –** SNPs that you found but are not in the mutation file
- **FN (False Negatives) –** SNPs that you should have found but did not

Look at the FP and FN list, in one paragraph discuss of some common aspects of the data that ended up on the FP and FN list.