Structural Variant (SV) calling with SVdetect 2021

Overview

Most approaches for predicting structural variants require you to have paired-end or mate-pair reads. They use the distribution of distances separating these reads to find outliers and also look at pairs with incorrect orientations. As mentioned during several of the presentations, many researchers choose to ignore these types of mutations and combined with the increased difficulty of accurately identifying them, the community is less settled on the "best" way to analyze them. Here we present a tutorial on a somewhat older program SVDetect. SVDetect is a type of program that makes use of configuration files rather than command line options (something you may encounter with other programs in your own work).

Other possible tools:

- BreakDancer - hard to install prerequisites on TACC. Requires installing libgd and the notoriously difficult GD Perl module.
- PEMer - hard to install prerequisites on TACC. Requires "ROOT" package.

Good discussion of some of the issues of predicting structural variation:

- http://www.genome.gov/Pages/Research/DER/1000GenomesProjectTutorials/StructuralVariants-JanKorbel.pdf

Comparison of many different SV tools

- https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1828-7#citeas

Learning objectives:

1. Identify structural variants in a new data set.
2. Work with a new type of program that uses configuration files rather than entering all information on a single command at the command line. This is very similar to the queue system TACC uses making this a good introduction.

Calling SV with SVdetect:

Here we'll look an E. coli genome re-sequencing sample where a key mutation producing a new structural variant was responsible for a new phenotype involving citrate, something the Barrick lab has studied.

Prepare your directories

suggested directory set up. Note the copy command must be run while on the head node, not an idev node

```
cds
cp -r $BI/gva_course/structural_variation/data GVA_sv_tutorial
cd GVA_sv_tutorial
```

This is Illumina mate-paired data (having a larger insert size than paired-end data) from genome re-sequencing of an *E. coli* clone.

File Name	Description	Sample
`61FTVAAXX_2_1.fastq`	Paired-end Illumina, First of mate-pair, FASTQ format	Re-sequenced *E. coli* genome
`61FTVAAXX_2_2.fastq`	Paired-end Illumina, Second of mate-pair, FASTQ format	Re-sequenced *E. coli* genome

`NC_012967.1.fasta`	Reference Genome in FASTA format	*E. coli* B strain REL606
NC_012967.1.lengths	Simple tab delimtered file based on the size of the reference needed for SVDetect so you don't have to create it yourself	

Map data using bowtie2

First we need to (surprise!) map the data. This will hopefully reinforce the bowtie2 tutorial you just completed.

> **Do not run on head node**
>
> Use **hostname** to verify you are still on the idev node.
>
> If not, and you need help getting a new idev node, see this tutorial.

```
bowtie2-build NC_012967.1.fasta NC_012967.1
bowtie2 -t -p 64 -X 5000 --rf -x NC_012967.1 -1 61FTVAAXX_2_1.fastq -2 61FTVAAXX_2_2.fastq -S 61FTVAAXX.sam
```

Possibly unfamiliar options:

- `--rf` tells bowtie2 that your read pairs are in the "reverse-forward" orientation of a mate-pair library
- `-X 5000` tells bowtie2 to not mark read pairs as discordant unless their insert size is greater than 5000 bases.

You may notice that these commands complete pretty quickly. Always remember speed is not necessarily representative of how taxing something is for TACC's head node, and always try to be a good TACC citizen and do as much as you can on idev nodes or as job submissions

Install SVDetect

This will be the most complicated installation yet. In addition to needing to install several different programs in the same conda installation command, we will need to install perl modules through cpan. Unfortunately, the cpan network can not be accessed through the compute nodes, so you must log out of your idev session using the logout command before continuing. If you are unsure if you are in an idev session remember you can use the **hostname** command to check.

conda installation

Like we saw in our samtools installation, we will need to install several programs at the same time to make sure they are all going to work with each other. In addition, we are going to create a new environment for working with SVDetect as some of the dependencies of SVDetect clash with those of samtools.

```
conda create --name SVDetect -c bioconda -c conda-forge -c imperial-college-research-computing _libgcc_mutex
perl libgcc-ng svdetect
```

We can now activate our new environment

```
conda activate SVDetect
```

cpan module installations

> **Again, make sure you are NOT on an idev node for working with cpan**

If you attempt to launch **cpan**, you likely get a message similar to the following:

```
/home1/0004/train402/miniconda3/envs/svdetect/bin/perl: symbol lookup error: /home1/apps/bioperl/1.007002/lib
/perl5/x86_64-linux-thread-multi/auto/version/vxs/vxs.so: undefined symbol: Perl_xs_apiversion_bootcheck
```

Using the "which -a" command shows us we actually have access to multiple different cpan executables. Friday's class will discuss more about how you can end up with multiple executable files named the same thing stored in different directories, and how the command line will treat them

```
which -a cpan
```

```
~/miniconda3/envs/SVDetect/bin/cpan
/bin/cpan
/usr/bin/cpan
```

While just typing **cpan** the first location was used and we saw it didn't work as we had hoped. We can explicitly launch the 2nd location by using the full path to the executable file on the prompt

```
/bin/cpan
```

In the following block note that each elipse will include large blocks of scrolling text as different modules are downloaded and installed. The process will take several minutes in total, just be ready to execute the next command when you get the cpan prompt back.

Install Perl modules required for SVDetect. (I do not know why the words do and for are appearing in bold, they are not meant as some kind of hint).

```
# choose 'yes' to do as much automatically as possible
# choose 'local::lib' for the approach you want (as you don't have admin rights on TACC)
# choose 'yes' to append the information to your .bashrc file
...
cpan[1]> install Config::General
...
cpan[2]> install Tie::IxHash
...
cpan[3]> install Parallel::ForkManager
...
cpan[4]> quit
```

How to fix cpan if downloads give errors

Several students were having trouble with the cpan downloads that seem to be related to some kind of interruption in the initial download process. The following commands have solved the issue for at least 1 student. Please try the following if you were unable to get the above cpan downloads to work, and let me know if you continue to experience difficulties.

relaunch cpan

```
/bin/cpan
```

On the cpan prompt

```
o conf init
```

This should then get you back to the point where you can:

Install Perl modules required for SVDetect. (I do not know why the words do and for are appearing in bold, they are not meant as some kind of hint).

```
# choose 'yes' to do as much automatically as possible
# choose 'local::lib' for the approach you want (as you don't have admin rights on TACC)
# choose 'yes' to append the information to your .bashrc file
...
cpan[1]> install Config::General
...
cpan[2]> install Tie::IxHash
...
cpan[3]> install Parallel::ForkManager
...
cpan[4]> quit
```

The above solution is based on steps 4-7 of this page. Again, if this does not fix the problem you were having, please let me know.

Why we choose "yes" to append information to our .bashrc file.

Just before getting your cpan prompt, there is a block of text that looks something like this

```
PATH="/home1/0004/train402/perl5/bin${PATH:+:${PATH}}"; export PATH;
PERL5LIB="/home1/0004/train402/perl5/lib/perl5${PERL5LIB:+:${PERL5LIB}}"; export PERL5LIB;
PERL_LOCAL_LIB_ROOT="/home1/0004/train402/perl5${PERL_LOCAL_LIB_ROOT:+:${PERL_LOCAL_LIB_ROOT}}"; export
PERL_LOCAL_LIB_ROOT;
PERL_MB_OPT="--install_base \"/home1/0004/train402/perl5\""; export PERL_MB_OPT;
PERL_MM_OPT="INSTALL_BASE=/home1/0004/train402/perl5"; export PERL_MM_OPT;

Would you like me to append that to /home1/0004/train402/.bashrc now? [yes]
```

The answer here is yes. What is being asked is if you would like the computer by default to be able to access these new perl 'lib' (libraries) you have created, and if you want perl binaries in your PATH. Recall that the PATH variable is where the command line searches when you enter a command so that you don't have to specify its location from the root directory. Similar is true of the PERL_LOCAL_LIB_ROOT and other variables except instead of being searched from the command line, the perl program searches them when commands inside the perl script are accessed. While there are ways to specify how to access these libraries when running individual commands, that is going to be much more complicated at a minimum, and may require editing scripts or programs.

Once you quit cpan, you will get a message to restart your shell. Since you are on a remote computer, you can accomplish the same thing by **log**ging **out** of TACC and **ssh**ing back in.

If the above bold letters are not enough of a clue for what you need to do here (and/or where you need to go to find appropriate minitutorials), now is a good time to start thinking about what question you need to be asking or sending in an email. It is ok to be overwhelmed or lost especially with the class being virtual and not being able to get good feedback from me directly on your progress. I am happy too help, but can only do so if I know you are struggling.

Once you have logged back in, be sure to restart a new idev session, and activate your SVDetect conda environment.

Analyze read mapping distribution

The first step is to look at all mapped read pairs and whittle down the list only to those that have an unusual insert sizes (distances between the two reads in a pair).

```
cd $SCRATCH/GVA_sv_tutorial
BAM_preprocessingPairs.pl -p 0 61FTVAAXX.sam
```

As we discussed in our earlier presentation, SV are often detected by looking for variations in library insert sizes. The stdout of the pearl script will answer the questions:

1. -- using -1142.566-5588.410 as normal range of insert size
2. Approximately 20% based on:

   -- 994952 mapped pairs

   ---- 195705 abnormal mapped pairs
3. Approximately 0.5% based on:

   -- Total : 1000000 pairs analysed

   -- 5048 pairs whose one or both reads are unmapped
4. Possible things are:

   a. The first answer can tell you what type of library it is if you did not know ahead of time (remember paired end reads have ~500-700bp inserts on average not 1000s of bp)
   b. This should help underscore that a significant portion of your total reads (and thus variation) may be in structural variants. Unfortunately, this does require generating a mate-pair library to learn.
   c. Low levels of unmapped read pairs suggests that both the reference is accurate, of a high quality, and free of contamination.
      i. Note that contamination in this case refers only to other organisms, and adapter sequences, **not other samples**.

Running SVDetect

SVDetect demonstrates a common strategy in some programs with complex input where instead of including a lot of options on the command line, it reads in a simple text file that sets all of the required options. Lets look at how to create a configuration file:

This is one of the most common mistakes people make during the course and the nature of zoom make this very likely to be true this year as well

Notice the next block contains line numbers. On lines 7 and 8 you see ##### and <USERNAME> ... these need to correspond to your scratch directory locations. You can easily check this with the pwd command.

Do not change anything else on the line.

If you are unsure what you should be replacing those place holders with please get my attention on zoom and I'll help you through it.

Create the file svdetect.conf with this text

```
<general>
input_format=sam
sv_type=all
mates_orientation=RF
read1_length=35
read2_length=35
mates_file=/scratch/#####/<USERNAME>/GVA_sv_tutorial/61FTVAAXX.ab.sam
cmap_file=/scratch/#####/<USERNAME>/GVA_sv_tutorial/NC_012967.1.lengths
num_threads=48
</general>

<detection>
split_mate_file=0
window_size=2000
step_length=1000
</detection>

<filtering>
split_link_file=0
nb_pairs_threshold=3
strand_filtering=1
</filtering>

<bed>
  <colorcode>
    255,0,0=1,4
    0,255,0=5,10
    0,0,255=11,100000
  </colorcode>
</bed>
```

The following commands will take a few minutes each and must be completed in order, so no advantages/ability to have them run in the background. Consult the manual for a full description of what these commands and options are doing while the commands are running.

Commands to run SNVDetect

```
SVDetect linking -conf svdetect.conf
SVDetect filtering -conf svdetect.conf
SVDetect links2SV -conf svdetect.conf
```

Take a look at the final output file: `61FTVAAXX.ab.sam.links.filtered.sv.txt`. Another downside of command line applications is that while you can print files to the screen, the formatting is not always the nicest. On the plus side in 95% of cases, you can directly copy the output from the terminal window to excel and make better sense of what the columns actually are

I've highlighted a few lines below:

```
chr_type        SV_type BAL_type        chromosome1     start1-end1     average_dist    chromosome2     start2-
end2     nb_pairs        score_strand_filtering  score_order_filtering   score_insert_size_filtering
final_score     breakpoint1_start1-end1 breakpoint2_start2-end2
...
INTRA   NORMAL_SENSE    -       chrNC_012967    599566-601025   -       chrNC_012967    663036-664898   430
100%    -       -       1       -       -
...
INTRA   NORMAL_SENSE    -       chrNC_012967    3-2025  -       chrNC_012967    4627019-4628998 288     100%
-       -       1       -       -
...
INTRA   REVERSE_SENSE   -       chrNC_012967    16999-19033     -       chrNC_012967    2775082-2777014 274
100%    -       -       1       -       -
```

1. This is a tandem head-to-tail duplication of the region from approximately 600000 to 663000.
2. This is just the origin of the circular chromosome, connecting its end to the beginning!
3. This is a big chromosomal inversion mediated by recombination between repeated IS elements in the genome. It would not have been detected if the insert size of the library wasn't > ~1,500 bp!

... Many of the others are due to new insertions of transposable elements.

Return to GVA2021 course page.