**SV workshop: How to call SV and compare them**

Welcome to the SV workshop. The following steps should give you insides on how to run SV calling over de novo assembly, short read mapping an long read mapping data. Clearly this is just a taste of the current field. If you for some reasons get lost over the exercise you can find backup data in ~/lecture\_SVCalling/results/ . Also a bash script is made available with all the commands: ~/lecture\_SVCalling/run.sh

**Please note that copy paste commands are sometimes tricky depending on what computer and interface you are using. If you encounter an error try to type the command.**

1. **Assembly based detection of SV**
   1. Change into the assembly directory:

cd ~/lecture\_SVCalling/examples/assembly

* 1. Align it using

*nucmer -maxmatch -l 100 -c 500 ../../Saccharomyces\_cerevisiae.R64-1-1.dna.toplevel.fa yeast\_pilon\_assembly.fasta*

* 1. Compress out.delta :

*gzip out.delta*

* 1. Download the out.delta.gz. Type in the IP assigned to you in the borwser and navigate to lecture\_SVCalling/examples/assembly
  2. Go to <http://assemblytics.com/> and upload the out.delta.gz
  3. What plots and summary statistics do you see?
  4. Copy the file

cp ~/lecture\_SVCalling/results/assembly/my\_favorite\_organism.Assemblytics\_structural\_variants.bed .

* 1. Convert this with SURVIVOR:

*SURVIVOR convertAssemblytics my\_favorite\_organism.Assemblytics\_structural\_variants.bed 0 assemblytics.vcf*

* 1. Run stats to see how many SV are called:

*SURVIVOR stats assemblytics.vcf -1 -1 -1 summary*

1. **Short read mapping based detection of SV**
   1. Change to folder:

cd ~/lecture\_SVCalling/examples/illumina

* 1. Now we are going to infer SVs using Delly: (watch the – they could be wrongly copied)

*delly call –o illumina\_mapped.delly –g ../../Saccharomyces\_cerevisiae.R64-1-1.dna.toplevel.fa illumina\_mapped.sort.*

* 1. Conver the output to VCF file:

*bcftools view illumina\_mapped.delly > illumina\_mapped.delly.vcf*

* 1. Run stats to see how many SV are called:

*SURVIVOR stats illumina\_mapped.delly.vcf -1 -1 -1 summary*

1. **Long read mapping based detection of SV**
   1. Change to folder:

cd ~/lecture\_SVCalling/examples/ont

* 1. Now we want to identify SVs using Sniffles:

sniffles -m Ont\_mapped.sort.bam -v Ont\_mapped.sniffles.vcf

* 1. Run stats to see how many SV are called:

*SURVIVOR stats* Ont\_mapped.sniffles.vcf *-1 -1 -1 summary*

1. **Comparison of the SV calls obtained**

As the last step today we want to compare the calls that we generated.

* 1. First we generate a file including the file names and paths.

cd ~/lecture\_SVCalling/examples/compare

*ls ../\*/\*vcf > file*

* 1. Next merge the vcf files and generate a new file using:

*SURVIVOR merge file 1000 1 1 0 0 0 comparison.vcf*

* 1. Open the file using: less -S *comparison.vcf*
  2. We want to identify the overlap:

*perl -ne 'print "$1\n" if /SUPP\_VEC=([^,;]+)/' comparison.vcf | sort | uniq -c*

* + 1. What does that command do?
  1. Why are there so many singeltons in the short read vs. long read caller?
     1. *grep 'SUPP\_VEC=010;' comparison.vcf | perl -ne 'print "$1\n" if /SVTYPE=([^,;]+)/' | sort | uniq -c*
     2. *grep 'SUPP\_VEC=001;' comparison.vcf | perl -ne 'print "$1\n" if /SVTYPE=([^,;]+)/' | sort | uniq -c*
     3. What do you see?