PsiSeq2_manual

Charlie Soeder

5/25/2018

Introduction

PsiSeq2 is an updated version of the software presented in [@Earley2011]. . . ..

Installation & Structure

Requirements:

Structure

PsiSeq2 is a SnakeMake pipeline consisting of a rule file (. . .) and a configuration file (. . .). The rule file defines the analytic workflow as discrete steps; these will be detailed later. The config file information such as paths to reference genomes, LiftOver chains, and executables; it also tabulates metadata about the sequenced reads to be analyzed.

default organization of sequenced reads are by treatment, then by sample: FASTQs/selection/SRR303333/SRR303333.fq

```
snakemake --restart-times 6 --latency-wait 60 --jobs 24 -p --cluster "sbatch --time={params.runtime} -n
```

Example Use Case

To illustrate the PsiSeq2 workflow, the data from [@Earley2011] will be reprocessed.

Acquiring and Aligning the Reads

Sequenced reads can be provided from experiment; read QA/QC is beyond the scope. If the required reads are not available, they will be generated using ART read simulator (if the config file specifies them as synthetic and lists a reference genome for fragmenting), or will be downloaded from NCBI (if the config file specifies them as synthetic and lists an SRA number).

In this case, the [@Earley2011] sequence from a backcrossed Sec/Sim line is downloaded from NCBI:

```
snakemake FASTQs/selection/SRR303333/SRR303333.fq
```

Synthetic reads are built from the reference genomes; here's the simulated reads from droSim1:

```
snakemake FASTQs/parent/SynthSec/SynthSec_1.fq FASTQs/parent/SynthSec/SynthSec_2.fq
```

Once the reads are available, the next step in the workflow is to align them to the two parental genomes. For example, this command will align the synthetic droSim1 reads against the droSec1 reference genome, using bwa as the aligner:

```
snakemake mapped_reads/droSim1/droSim1_vs_droSec1.bwa.sort.bam
```

SnakeMake pipelines are recursive; calling the above command will also check for its dependencies (the synthetic reads), and it will run the read simulator if it doesn't find them.

From the alignment, a pileup file is generated:

```
snakemake mapped_reads/droSim1/droSim1_vs_droSec1.bwa.mpileup
```

Shared SNPs files:

Once pileups have been generated for the trio, the offspring pileup is compared to both of the parent pileups:

```
snakemake -p variant_comparisons/SRR303333_and_SynthSim_vs_droSec1.bwa.sharedSnps.bed SRR303333_and_Syn
```

These are equivalent to the output of shared_snps_v3.pl in [@Earley2011]. They are mapped to a common consolidated coordinate system using USCS LiftOver:

```
snakemake -p variant_comparisons/SRR303333_and_SynthSim_vs_droSec1.bwa.lift2dm6.sharedSnps.bed SRR30333
```

*counting (by window, by bin)

Once the SNPs have been compared between an offspring and parent, they are grouped and counted. This can be done using genomic windows of fixed width; to use windows 100kb wide, sliding by 10kb:

```
snakemake analysis_out/SRR303333_and_SynthSim_vs_droSec1.bwa.lift2dm6.dm6_w100000_s10000.windowCounts.b
```

This will output a BED file listing the windows and counting the SNPs observed in the parent (column 5) and of those, the count observed in the offspring (column 4)

[@Earley2011] implemented this slightly differently, binning by a fixed number of SNPs rather than by a fixed window size. This can be implemented as well. For example, to bin 1000 SNPs at a time, sliding by 250 SNPs:

```
snakemake analysis_out/SRR303333_and_SynthSim_vs_droSec1.bwa.lift2dm6.b1000s250.snpBins.bed
```

This outputs a BED file listing the genome intervals corresponding to the bins, as well as the number of sites in each bin which are variable in the offspring.

*freebayes

*wrap it up (Build This File)

The output analysis can be used downstream, or visualized as in this PDF. The desired output can be summoned recursively by specifying it as a requirement for a summary rule; for example, the data graphed in this PDF was included in the input for the final PDF-building rule:

```
rule build_PsiSeq2_manual:
    input:
        window_counts = ["analysis_out/SRR303333_and_SynthSim_vs_droSec1.bwa.lift2dm6.dm6_w100000_s1000
        bin_counts = ["analysis_out/SRR303333_and_SynthSim_vs_droSec1.bwa.lift2dm6.dm6_w100000_s10000.w
#       freebayes_counts = []
    params:
        runmem_gb=8,
        runtime="1:00:00"
    output:
        pdf_report = "PsiSeq2_Manual.pdf"
    run:
        shell("sh scripts/markerDown.sh scripts/PsiSeq2_Manual.Rmd {output.pdf_report}")
```

```
snakemake -s manualBuilder.snakefile PsiSeq2_Manual.pdf
```