contig_assembly

March 14, 2018

1 Contig assembly

After cleaning and trimming the reads in the previous step we are now ready to use the fastq-reads for de-novo contig assembly. In this step the overlap between fastq reads is being used to build long, uninterrupted sequences, among which we hopefully find the target regions that were selected for during sequence capture. In a way a contig can be seen as a consensus of several reads:

However, the underlying algorithms for building contigs are much more complex than in the simplified image above. In our pipeline you can choose between the assembly programs ABySS and Trinity. We tested both assemblers and they both appear to produce very similar results. If you are working with sequence capture data of DNA regions (exons, introns, UCEs, mitochondrial markers, etc.) we recommend you to use ABySS since this assembler was built for assembling DNA sequences. Trinity on the other hand was built for assembling transcriptome (RNA) sequences.

1.1 Running the assembly

For the assembly step we use the `assemble_reads` function of secapr. For an overview of all available flags execute the help function:

```
In [2]: %%bash
        secapr assemble_reads -h
```

```
usage: secapr assemble_reads [-h] --input INPUT --output OUTPUT
                             [--assembler {trinity,abyss}] [--kmer KMER]
                             [--contig_length CONTIG_LENGTH] [--single_reads]
                             [--cores CORES]

Assemble trimmed Illumina read files (fastq)

optional arguments:
  -h, --help            show this help message and exit
  --input INPUT         Call the folder that contains the trimmed reads,
                        organized in a separate subfolder for each sample. The
                        name of the subfolder has to start with the sample
                        name, delimited with an underscore [_]
  --output OUTPUT       The output directory where results will be saved
```

```
  --assembler {trinity,abyss}
                        The assembler to use.
  --kmer KMER           Set the kmer value
  --contig_length CONTIG_LENGTH
                        Set the minimum contig length for Trinity assembly.
                        Contigs that are shorter than this threshold will be
                        discarded.
  --single_reads        Use this flag if you additionally want to use single
                        reads for the assembly
  --cores CORES         For parallel processing you can set the number of
                        cores you want to run Trinity on.
```

Now we run the assembly with the default options, just like this:

```
secapr assemble_reads --input ../../data/processed/cleaned_trimmed_reads/ --output
```

The assembly step is very time intensive and may take several hours or even days, depending on the number of samples and the size of the files. For our example dataset the assembly took approximately 45 min per sample. The assembly step produces a fasta file for each sample, containing all assembled contig sequences. There are commonly 1000s or even 100,000s of sequences in the contig fasta file, many of which represent random short sequences that were present during sequencing. You may also find some very long sequences in the sample file which may represent the mitochondrial genome or in some cases big parts of the chloroplast genome (in plants). In the case of sequence capture datasets, we are mostly interested in the contigs that represent our enriched target sequences. We will show you in the next step how these can be easily extracted from the contig file by using a reference fasta file containing templates for the sequences of interest (often the file used to design the RNA baits). Go to the manual for extracting target contigs.

Previous page | Next page

```
In [ ]:
```