IN-BIOSx000 2018 exam: assembly

Before the home exam date, you have to solve the tasks described here, and prepare a 15 to 20-minute presentation to be held on the exam date. The following contains explanations of the tasks, and directions for what to present.

Along the way, I will pose questions you need to present to answer to during your presentation. You have around 20 minutes, so don't try to put too much on your slides. I'd rather see some well-presented interesting results, than many slides with too little time for the last ones. Please also present, and explain, the actual command lines for the steps you performed.
Do not submit answers in writing.

NOTE: some tasks are for PhD students only!

If you have a pressing question or a problem that stops you from completing the tasks, contact me at karin.lagesen@vetinst.no

Make sure you read the instructions carefully!

Background

You will be using datasets quite similar to the ones we used during the course, but from a different bacterial species: ***Meiothermus ruber* DSM 1279**. The data was used for a few publications:

- Tindall *et al.*, "Complete genome sequence of Meiothermus ruber type strain (21)" Stand Genomic Sci 3(1): 26-36 (2010). DOI: [10.4056/sigs.1032748](http://dx.doi.org/10.4056/sigs.1032748)
- Illumina Application Note "De Novo Assembly of Bacterial Genomes", <http://www.illumina.com/content/dam/illumina-marketing/documents/products/appnotes/appnote-nextera-mate-pair-bacteria.pdf>
  ● Chin et al, Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nature Methods 10, 563–569 (2013). DOI: [doi:10.1038/nmeth.2474](http://dx.doi.org/doi:10.1038/nmeth.2474)

There is no requirement to read these, nor will it help for the exam, but you're obviously welcome to do so.

Please note that the genome of this strain is 3.1Mbp, not 4.6 Mbp!

Data

The input data is located here:

/share/inf-biox121/data/exam/assembly

The following files are provided:

- in the `illumina` folder`
  - SRR1800541_R1_50x.fastq` and `SRR1800541_R2_50x.fastq` contain reads from a HiSeq run of the strain subsampled to 50x coverage.
  - `SRR1800541_R1_350x.fastq` and `SRR1800541_R1_350x.fastq` contain reads from a HiSeq run of the strain subsampled to 350x coverage.
  - `ERR76053X_MP_R1_trimmed.fastq` and `ERR76053X_MP_R2_trimmed.fastq` contain the mate pair data from a MiSeq run, trimmed for adaptors and low-quality bases
- in the `pacbio` folder
  - `SRR8118XX_m120803.filtered_subreads.fastq` contains 124x coverage of slightly older PacBio reads (so-called `C2XL` chemistry) in fastq file format (these are the so-called subreads).
  - the folders `SRR811863`, `SRR811863` and `SRR811865` contain the raw data from these pacbio runs
- in the `reference` folder
  - `CP005385.1_Meiothermus_ruber_DSM_1279.fna` contains the genome sequence in `fasta` format
  - `CP005385.1_Meiothermus_ruber_DSM_1279.gff` the annotated genes in `gff` format

NOTE there are no MinION data available for this strain.

All tasks

- always use the `assemblathon_stats.pl` script to judge assemblies
- **IMPORTANT** use a minimum gap length of 1 bp for all analysis (not the default from each tool). This affects
  - `assemblathon_stats.pl `
  - `scaffoldgap2bed.py`
- show the commands you used
- for each 'final' assembly, show the most relevant metrics.

Part 1: Assembly

Task 1.1: velvet assembly of the Illumina paired end reads only

- use the 50x data for this task
- determine optimal `kmer` size for an assembly with **the paired end reads only**
- to find the optimal `kmer`, do not use the pairing information or `-cov_cutoff` or `-exp_cov` parameters
- using your obtained optimal `kmer` size, perform a default velvet assembly, this time using the pairing information in the reads, with auto set for exp_cov and cov_cutoff

Call this assembly `velvet_pe`

Questions to answer:

- show a table of each `kmer` with their N50, longest contig and total assembly length
- which is the "best" kmer for this data using velvet?
- how much did adding the pairing information help for the assembly quality?
- what was the estimated values for exp_cov and cov_cutoff?

Task 1.2: velvet assembly of the illumina paired end reads and mate pair reads

- create a velvet assembly using PE and MP reads
- use the `best kmer` size as found in the task above, add the mate pair reads for a new velvet assembly
- the mate pair reads are already in the orientation that velvet expects them to be (so-called 'forward-reverse')
- use auto for exp_cov and cov_cutoff
- use the `node_coverage.ipynb` jupyter notebook on the velvet `stats.txt` file from this assembly and show the plot.

Call this `velvet_pe+mp`

Questions to answer:

- what did velvet determine to be the insert size for both libraries, and what was the standard deviation?
- what values did velvet set for exp_cov and cov_cutoff?
- judging from the notebook figure, do you agree with the values set for exp_cov and cov_cutoff by velvet?

Task 1.3: SPADES Illumina-only assembly

- use the **350x data** for this task (!)
- perform the SPADES assembly as described for the course with the exam data (both paired end and mate pair)
- settings to think about:

    - `-t` number of CPUs to use: do not use more than 3
    - `--memory` memory usage, don't use more than 30
    - `-k` k-mer range: use `21,29,37,43`
    - `--mp1-fr` this tels SPADES the orientation of the mate pairs

Call this assembly `spades_pe+mp`

Questions to answer:
- how do the assembly stats differences between the velvet and spades assemblies?
- what would you think is the reason for the differences?

Task 1.4: PHD student only! SPADES hybrid assembly
- use the **350x data** for this task (!)
- replace the mate pair data with the PacBio reads (the `fastq` file with subreads) as done during the course
- run the assembly the same as for the previous task
- call this `spades_hybrid`

Questions to answer:
- did replacing the mate-pair data with pacbio reads improve the assembly,
- if so, in what way?
- if so, why and how did the pacbio reads improve the assembly?

Task 1.5: PacBio only assembly using canu
- perform the assembly as for the course, using the `fastq` file with subreads
- call this `canu_pacbio`

Questions to answer:
- how does the canu assembly stats compare to those from SPADES?

Part 2: reference free assembly evaluation

For this part, use the following assemblies:
- `velvet_pe+mp`, the paired end and mate pair velvet assembly
- `spades_pe+mp`, the Illumina-only SPADES assembly

Task 2.1: Mapping
- map the paired end reads and mate pair reads to the scaffold files with BWA

Task 2.2: REAPR
- perform REAPR analysis for the assemblies, as per the course instructions
- put the REAPR 05.summary.report.txt contents for each assembly in a table and show it

Task 2.3: Genome browser

- add the following to IGV genome browser for both assemblies:
    - assembled scaffolds
    - gap track (use `scaffoldgap2bed.py` NOTE 1 bp minimum gap size)
    - mapped reads
    - Reapr's `03.score.errors_nospaces.gff` file (after converting spaces to underscores as described in the tutorial)

Questions to answer:
- for each assembly, how many places did REAPR decide to make a break?
- for the `velvet_pe+mp` assembly, show one or two screenshots from IGV of alignments from positions where REAPR decided to split a scaffold/contig. Explain why you agree/disagree with REAPR?

Part 3: reference-based assembly evaluation

For this part, use the following assemblies:
- `velvet_pe+mp`, the paired end and mate pair velvet assembly
- `spades_pe+mp`, the Illumina-only SPADES assembly
- **PhD students only** `spades_hybrid`, the hybrid SPADES assembly
- `canu_pacbio`, the PacBio canu assembly

Task 3.1: Reference comparison

Compare the assemblies to the reference sequence using `quast`. You will need the fna and the gff files mentioned above.
- show the results files during the exam (both the extended report and the contig viewer).

Questions you need to answer
- was one assembly clearly “better” than the others when compared to the reference?
- if so, how and in what way?
- given all of the above, which assembly is “better”?

Good luck!