Racer on HPC

What is Racer?

RACER (Rapid and Accurate Correction of Errors in Reads) is a software tool for correcting errors in DNA sequencing reads. It is designed to handle high error rates from long-read sequencing technologies, such as Oxford Nanopore, and it can be used to improve the quality of downstream analysis results.

RACER uses a k-mer based approach for error correction, where it identifies and corrects errors based on the frequency of overlapping k-mers. It also implements a statistical framework for assessing the confidence of correction decisions, which helps to avoid correcting correct bases.

RACER has been shown to provide accurate error correction with high precision and recall, making it a useful tool for a wide range of genomic applications. It is also designed to be fast and scalable, making it well-suited for large datasets.

Links:

Academic Paper

GitHub

Versions Available:

The following versions are available on the cluster:

- Racer

How to load Racer?

To load Racer, use the following commands:

```
#Load the Racer module
module load bio/racer
```

To verify if the module is loaded correctly, use the following command,

```
# List all the module loaded in the environment
module list
```

In a fresh environment, this only load Racer module without any dependencies.

How to use Racer?

Prepare input data by converting your raw sequencing reads into a format that RACER can use as input. The following are the general steps involved in preparing the input data:

1. Formatting: Ensure that your raw sequencing reads are in a supported format, such as FASTQ or BAM. If necessary, use a tool such as SAMtools to convert the reads into the correct format.
2. Quality control: Perform quality control on the sequencing reads to check for any issues that could impact the accuracy of the error correction. This may involve trimming low-quality reads, removing adapters, and filtering out reads with high levels of duplicates.

To run RACER use the command:

```
< racer > < inputReads > < correctedReads > < genomeLength >
```

where

- < racer > is the executable
- < inputReads > is the input file containin the reads; fasta or fastq
- < correctedReads > will contain the corrected reads at the end
- < genomeLength > is the approximate length of the DNA molecule that originated the reads, such as the genome length in a whole genome sequencing project

- -if only parts of a genome were sequenced, then only the total length of those parts should be used (instead of the length of the total genome)
- -precise value is not necessary, an approximation will work well

Here is a sample SLURM script to run racer in HPC,

```
#!/bin/bash
#SBATCH --job-name=racer
#SBATCH --output=racer.out
#SBATCH --error=racer.err
#SBATCH --time=48:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
#SBATCH --mem=64GB
 #SBATCH --p main
#SBATCH --qos main

# Load RACER module
module load bio/racer

# Define the input and output files
reads=input.fastq
corrected_reads=output.fastq

# Run RACER
racer  --input $reads --output $corrected_reads

# Check the exit status of the RACER job
if [ $? -ne 0 ]; then
    echo "Error: RACER job failed"
    exit 1
fi

# Exit the SLURM script
exit 0
```

Where to find help?

If you are confused or need help at any point, please contact OIT at the following address.

https://ua-app01.ua.edu/researchComputingPortal/public/oitHelp