Oct 25, 2022

Mitochondrial genome assembly

Graham Etherington[1]

[1]The Earlham Institute

1 *Works for me* Share

dx.doi.org/10.17504/protocols.io.bqzbmx2n

Graham Etherington
The Earlham Institute

ABSTRACT

De novo assembly of 49 mustelid whole mitochondrial genomes

DOI

dx.doi.org/10.17504/protocols.io.bqzbmx2n

EXTERNAL LINK

https://doi.org/10.1093/jhered/esac038

PROTOCOL CITATION

Graham Etherington 2022. Mitochondrial genome assembly. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bqzbmx2n

MANUSCRIPT CITATION please remember to cite the following publication along with this protocol

Etherington GJ, Ciezarek A, Shaw R, Michaux J, Croose E, Haerty W, Palma FD, Extensive genome introgression between domestic ferret and European polecat during population recovery in Great Britain. Journal of Heredity 113(5). doi: 10.1093/jhered/esac038

LICENSE

CREATED

Dec 23, 2020

LAST MODIFIED

Oct 25, 2022

PROTOCOL INTEGER ID

45827

1 Calculate read length of fastq files for each sample and run MitoZ

```
R1=$1
R2=$2

fq1="$(realpath $R1)"
fq2="$(realpath $R2)"

dir="$(dirname $R1)"
fpath="$(basename $R1)"
#get the sample name and the prefix for the output R1 and R2 reads
samplename="$(cut -d'_' -f1 <<< $fpath)"
#get the length of the 11th read (in case the first few are a bit
short). Method will vary depending if the reads are gzipped or not

extension="${fpath##*.}"
fastq_length=150
if [ $extension == "gz" ]; then
         fastq_length="$(zcat $fq1 | head -n 42 | sed -n '42p'  |
wc -c)"
 else
         fastq_length="$(sed -n '42p' $fq1 | wc -c)"
 fi

source samtools-1.10
source mitoz-2.3
source ncbiblast-2.2.27

REF=NC_020638.1_mitochondrial.fasta
#run mitzo all
srun mitoz all --fastq1 $fq1 --fastq2 $fq2 --fastq_read_length
$fastq_length  --outprefix $samplename --thread_number 16 --clade
Chordata --genetic_code 2 --filter_taxa_method 1
#re-order assembly so all are anchored to a common reference.
srun python
/ei/software/testing/mitoz/2.3/src/release_MitoZ_v2.3/useful_script
s/Mitogenome_reorder.py -f $samplename.result/work71.mitogenome.fa
-r $REF
```

2 Genome alignment. Concatenate the genomes and use ClustalW to align them.

Citation: Graham Etherington Mitochondrial genome assembly https://dx.doi.org/10.17504/protocols.io.bqzbmx2n

2.1 Rename both the accession name and file name of the genome assemblies, as they'll all have the same name (work71.mitogenome.fa.reorder)

```
SAMPLE=$1 #the sample name
FASTA=$2 #the path to the assembly

dir="$(dirname $FASTA)"

#change any number of upper and lowercase characters,
numbers, spaces and = sign to the sampleID
sed -i "s/>[A-Za-z0-9 =]*/>${SAMPLE}/g" "$FASTA"

#rename the fasta file from
'work71.mitogenome.fa.reorder' to e.g.
'euro_S01_mitogenome.fasta'
mv $FASTA $dir/$SAMPLE\_mitogenome.fasta
```

2.2 Concatenate all of the assemblies

```
find . -name "*_mitogenome.fasta" -exec cat {} \; -
printf "\n" > all_mtdna_genomes.fasta
```

You may need to visualise the genomes to make sure they're all the same complement. Reverse complement any genomes as required.

2.3 Align the assemblies and change format to FASTA

```
source clustalw-2.1
source emboss-6.6.0

#align the seqences
srun clustalw -ALIGN -INFILE=all_mtdna_genomes.fasta -
TYPE=DNA -OUTFILE=all_mtdna_genomes_aligned.aln
#reformat to fasta
srun seqret -sequence all_mtdna_genomes_aligned.aln -
outseq all_mtdna_genomes_aligned.fasta
```

Citation: Graham Etherington Mitochondrial genome assembly https://dx.doi.org/10.17504/protocols.io.bqzbmx2n