FEB 19, 2024

DOI:
dx.doi.org/10.17504/protocols.io.n2bvj37dnlk5/v1

Protocol Citation: Vidya Niranjan, Lavanya C, Shri Ganapathi, Spoorthi R Kulkarni 2024. Consensus Sequence Generation Protocol: Leveraging Variant Analysis in Mulberry Genotypes. **protocols.io** https://dx.doi.org/10.17504/protocols.io.n2bvj37dnlk5/v1

Consensus Sequence Generation Protocol: Leveraging Variant Analysis in Mulberry Genotypes

Vidya Niranjan[1], Lavanya C[1], Shri Ganapathi[2], Spoorthi R Kulkarni[1]

[1]R V College of Engineering; [2]R V college of Engineering

Centre of Excellence in Computational Genomics (Vidya Lab)

Vidya Niranjan

ABSTRACT

This research investigates four distinct Mulberry genotypes, each offering valuable traits essential for sericulture and broader agricultural sustainability. Notably, Thailand Male is examined for genetic variations associated with superior leaf yield, potentially informing selective breeding efforts for cultivars providing optimal nutrition to silkworms. Assam Bol's resistance to root rot diseases is explored to elucidate genetic factors, promising the development of disease-resistant Mulberry varieties and reducing reliance on chemical interventions. The drought-resistant S1 genotype's genetic resilience is scrutinized to develop Mulberry cultivars capable of withstanding water scarcity amid climate change challenges. Furthermore, the study assesses Punjab Local's nitrogen use efficiency, crucial for enhancing agricultural productivity while mitigating environmental impact. Employing advanced genetic analysis techniques, including SNP and SSR markers, this research aims to revolutionize Mulberry agriculture by identifying genetic markers, establishing linkage maps, and creating diagnostic tools. Ultimately, this endeavor seeks to enhance economic viability and environmental sustainability within the sericulture industry.
The current protocol is designed for development of consensus for mulberry genome. But the protocol can be used for other species too.

GUIDELINES

Kindly use the current commands in Ubuntu 20.04

Protocol status: Working
We use this protocol and it's working

Created: Feb 15, 2024

Last Modified: Feb 19, 2024

PROTOCOL integer ID: 95263

BEFORE START INSTRUCTIONS

Scripting and Command line operation should be known
The Server with higher configuration to be used for the protocol

The GATK has some filter values which will change according to the sample. The User has to set the values accordingly.
Java file has to be installed properly

SAMPLE COLLECTION AND REFERENCE GENOME

1 Collection of the Whole genome sequences and the Reference genome for *Morus Indica (Mulberry)*

The high-depth sequencing of multiple mulberry accessions were sequenced via Illumina platform and the fastq files *were retrieved from MindGP an Indian repository of Mulberry data.*. The project involves generation of a high-density SNP map for various genotyping applications.

Dataset

Dataset Collection from MINDGP NAME

http://tgsbl.jnu.ac.in/MindGP/raw_data.php LINK

Dataset

Dataset reposited in NCBI SRA NAME

https://www.ncbi.nlm.nih.gov/sra/?term=PRJNA728807 LINK

Command

Downloading the Genome sequences from FTP link in Ubuntu 20.04

```
wget <FTP Link>
```

2 Reference genome

The Morus indica species have 365 sequenced contigs which were combined together to get a whole reference genome.
The concatenated file was used for alignment and mapping of the fastq files and was named as mulberry_reference.fasta

Dataset

Mulberry Reference NAME

http://tgsbl.jnu.ac.in/MindGP/download/JNU_Mind_v1.2.fasta.gz LINK

ALIGNMENT AND MAPPING

3 The alignment and mapping process begins by indexing the reference genome for efficient alignment using BWA MEM. The aligned sequences are then stored in a SAM file format. Next, SAMtools is employed to convert the SAM file to the binary BAM format and to sort the aligned reads based on their genomic coordinates. This sorting step optimizes downstream analyses such as variant calling and consensus sequence generation.

Command

The command bwa mem mulberry_reference.fasta <SRRID>_1.fastq.gz <SRRID>_2.fastq.gz > <SRRID>.sam aligns paired-end sequencing reads from <SRRID>_1.fastq.gz and <SRRID>_2.fastq.gz to the reference genome mulberry_reference.fasta using BWA MEM. The alignment results are saved in the SAM file <SRRID>.sam.

```
bwa mem mulberry_reference.fasta <SRRID>_1.fastq.gz <SRRID>_2.fastq.gz
> <SRRID>.sam
```

4

Command

The command samtools view -bS <SRRID>.sam > <SRRID>.bam converts the SAM file <SRRID>.sam to the binary BAM format, saving the converted file as <SRRID>.bam.

```
samtools view -bS <SRRID>.sam > <SRRID>.bam
```

Command

The command samtools sort <SRRID>.bam -o <SRRID>_sorted.bam sorts the BAM file <SRRID>.bam and saves the sorted output as <SRRID>_sorted.bam.

```
samtools sort <SRRID>.bam -o <SRRID>_sorted.bam
```

VARIANT CALLING USING GATK PIPELINE

5 The GATK pipeline processes raw sequencing data by initially aligning reads to a reference genome and then recalibrating base quality scores to improve accuracy. Variant calling algorithms identify potential variants, including SNPs, which are subsequently filtered for high quality using variant quality score recalibration. Filtered variants are annotated with functional information before a consensus sequence is generated by incorporating the most common variant at each genomic position across all samples. This approach ensures reliable SNP identification, filtering, and consensus sequence generation.

6 The provided command is using the Picard tools, specifically the MarkDuplicates tool, in a Java environment. Let's break down the command:

7 java -jar picard.jar: This part of the command indicates that you are running a Java application and specifying the JAR (Java Archive) file picard.jar, which contains the Picard tools.
MarkDuplicates: This is the specific Picard tool being invoked. It is used to identify and mark duplicate reads in a BAM (Binary Alignment/Map) file.
INPUT=<SRRID>_sorted.bam: Specifies the input BAM file with aligned reads. <SRRID>_sorted.bam is a placeholder where <SRRID> represents the actual sample or experiment identifier. You would replace <SRRID> with the specific identifier for your data.
OUTPUT=<SRRID>_dedup_reads.bam: Specifies the output BAM file where the deduplicated reads will be written. <SRRID>_dedup_reads.bam is a placeholder for the deduplicated file, and the actual filename will be determined by the provided sample or experiment identifier.
METRICS_FILE=<SRRID>_metrics.txt: Specifies the file where metrics about the duplication process will be written. The metrics include information about the number of duplicates, optical duplicates, and other statistics. <SRRID>_metrics.txt is a placeholder for the metrics file, and the actual filename will be determined by the provided sample or experiment identifier.

Command

```
java -jar picard.jar MarkDuplicates INPUT=<SRRID>_sorted.bam OUTPUT=
<SRRID>_dedup_reads.bam METRICS_FILE=<SRRID>_metrics.txt
```

8 CollectAlignmentSummaryMetrics tool, to collect alignment summary metrics from a BAM (Binary Alignment/Map) file. Let's break down the command:

java -jar picard.jar: This part of the command indicates that you are running a Java application and specifying the JAR (Java Archive) file picard.jar, which contains the Picard tools.
CollectAlignmentSummaryMetrics: This is the specific Picard tool being invoked. It is used to collect summary metrics about the alignment from a BAM file.
R=mulberry_reference.fasta: Specifies the reference genome in FASTA format against which the reads were aligned. mulberry_reference.fasta is a placeholder, and you would replace it with the actual filename of your reference genome.
I=<SRRID>_dedup_reads.bam: Specifies the input BAM file containing deduplicated reads. <SRRID>_dedup_reads.bam is a placeholder, and you would replace <SRRID> with the specific identifier for your sample or experiment.
O=<SRRID>_alignment_metrics.txt: Specifies the output file where alignment summary metrics will be written. <SRRID>_alignment_metrics.txt is a placeholder, and the actual filename will be determined by the provided sample or experiment identifier.

Command

```
java -jar picard.jar CollectAlignmentSummaryMetrics R=mulberry_reference.fasta I=<SRRID>_dedup_reads.bam O=<SRRID>_alignment_metrics.txt
```

9

Command

```
java -jar picard.jar CollectInsertSizeMetrics -I <SRRID>_dedup_reads.bam -O <SRRID>_insert_metrics.txt -H <SRRID>_insert_size_histogram.pdf
```

10 AddOrReplaceReadGroups adds or replaces read groups in a BAM (Binary Alignment/Map) file. Read groups are metadata associated with sequencing data and are important for downstream analysis. Let's break down the command:

java -jar picard.jar: This part of the command indicates that you are running a Java application and specifying the JAR (Java Archive) file picard.jar, which contains the Picard tools.
AddOrReplaceReadGroups: This is the specific Picard tool being invoked. It is used to add or replace read group information in a BAM file.
I=<SRRID>_dedup_reads.bam: Specifies the input BAM file containing deduplicated reads. <SRRID>_dedup_reads.bam is a placeholder, and you would replace <SRRID> with the specific identifier for your sample or experiment.
O=<SRRID>_sorted_dedupreads.bam: Specifies the output BAM file where the reads with added or replaced read groups will be written. <SRRID>_sorted_dedupreads.bam is a placeholder, and the actual filename will be determined by the provided sample or experiment identifier.
RGID=4: Specifies the Read Group ID. It is a unique identifier for the read group. In this case, it's set to 4.
RGLB=lib1: Specifies the Read Group Library. It represents the library from which the sequencing data originates.
RGPL=ILLUMINA: Specifies the Read Group Platform. It indicates the platform used for sequencing, and here it's set to ILLUMINA.
RGPU=unit1: Specifies the Read Group Platform Unit. It provides additional information about the flowcell or lane, and here it's set to unit1.
RGSM=20: Specifies the Read Group Sample Name. It represents the sample name associated with the read group. Here, it's set to 20.

Command

```
java -jar picard.jar AddOrReplaceReadGroups I=<SRRID>_dedup_reads.bam O=
<SRRID>_sorted_dedupreads.bam RGID=4 RGLB=lib1 RGPL=ILLUMINA RGPU=unit1 RGSM=20
```

11

Command

INDEXING THE BAM FILE

```
samtools index <SRRID>_sorted_dedupreads.bam
```

12 Now that the Preprocessing is been performed, The GATK Pipeline is run to get the indels and the SNPs

Command

GATK PHASE 1

```
gatk HaplotypeCaller -R mulberry_reference.fasta -I
<SRRID>_sorted_dedupreads.bam -O <SRRID>_raw_variants.vcf

gatk SelectVariants -R mulberry_reference.fasta -V <SRRID>_raw_variants.vcf --
select-type-to-include SNP -O <SRRID>_raw_snps.vcf

gatk SelectVariants -R mulberry_reference.fasta -V <SRRID>_raw_variants.vcf --
select-type-to-include INDEL -O <SRRID>_raw_indels.vcf

gatk VariantFiltration -R mulberry_reference.fasta -V <SRRID>_raw_snps.vcf -O
<SRRID>_filtered_snps.vcf -filter-name "QD_filter" -filter "QD < 2.0" -filter-
name "FS_filter" -filter "FS > 60.0" -filter-name "MQ_filter" -filter "MQ <
40.0" -filter-name "SOR_filter" -filter "SOR > 4.0" -filter-name
"MQRankSum_filter" -filter "MQRankSum < -12.5" -filter-name
"ReadPosRankSum_filter" -filter "ReadPosRankSum < -8.0"

gatk VariantFiltration -R mulberry_reference.fasta -V <SRRID>_raw_indels.vcf -O
<SRRID>_filtered_indels.vcf -filter-name "QD_filter" -filter "QD < 2.0" -filter-
name "FS_filter" -filter "FS > 200.0" -filter-name "SOR_filter" -filter "SOR >
10.0"

gatk SelectVariants --exclude-filtered -V <SRRID>_filtered_snps.vcf -O
<SRRID>_bqsr_snps.vcf

gatk SelectVariants --exclude-filtered -V <SRRID>_filtered_indels.vcf -O
<SRRID>_bqsr_indels.vcf
```

Command

GATK PHASE 2

```
gatk BaseRecalibrator -R mulberry_reference.fasta -I
<SRRID>_sorted_dedupreads.bam --known-sites <SRRID>_bqsr_snps.vcf --known-sites
<SRRID>_bqsr_indels.vcf -O <SRRID>_recal_data.table

gatk ApplyBQSR -mulberry_reference.fasta -I <SRRID>_sorted_dedupreads.bam -bqsr
<SRRID>_recal_data.table -O <SRRID>_recal_reads.bam

gatk BaseRecalibrator -mulberry_reference.fasta -I <SRRID>_recal_reads.bam --
known-sites <SRRID>_bqsr_snps.vcf --known-sites <SRRID>_bqsr_indels.vcf -O
<SRRID>_post_recal_data.table

gatk AnalyzeCovariates -before <SRRID>_recal_data.table -after
<SRRID>_post_recal_data.table -plots <SRRID>_recalibration_plots.pdf
```

Command

GATK PHASE 3

```
gatk HaplotypeCaller - mulberry_reference.fasta -I <SRRID>_recal_reads.bam -O <SRRID>_raw_variants_recal.vcf

gatk SelectVariants -mulberry_reference.fasta -V <SRRID>_raw_variants_recal.vcf --select-type-to-include SNP -O <SRRID>_raw_snps_recal.vcf

gatk VariantFiltration -mulberry_reference.fasta -V <SRRID>_raw_snps_recal.vcf -O <SRRID>_filtered_snps_final.vcf -filter-name "QD_filter" -filter "QD < 2.0" -filter-name "FS_filter" -filter "FS > 60.0" -filter-name "MQ_filter" -filter "MQ < 40.0" -filter-name "SOR_filter" -filter "SOR > 4.0" -filter-name "MQRankSum_filter" -filter "MQRankSum < -12.5" -filter-name "ReadPosRankSum_filter" -filter "ReadPosRankSum < -8.0"
```

CONSENSUS GENERATION

13 The process combines information from multiple variants to generate a single consensus sequence. The consensus sequence provides a comprehensive summary of the genetic variation present in the samples and serves as a reference for downstream analysis, such as functional annotation or comparative genomics.

Command

bgzip compresses files in the Bgzip format, commonly used for genomic data like VCF files. It produces compressed files with the .gz extension, facilitating efficient storage and transmission of large datasets while maintaining random access to individual blocks within the compressed file.

```
bgzip <SRRID>_filtered_snps_final.vcf
```

Command

Tabix indexes VCF files to enable efficient retrieval of variant records based on genomic coordinates. This indexing enhances the speed of accessing specific genomic regions within large VCF files

```
tabix -p vcf <SRRID>_filtered_snps_final.vcf.gz
```

Command

samtools faidx creates an index for FASTA reference sequences, facilitating rapid retrieval of specific regions. bcftools consensus utilizes variant calls from a VCF file to generate a consensus sequence, integrating detected variants into the indexed reference sequence. These tools are essential for genomic analyses, allowing for efficient sequence retrieval and consensus sequence generation based on variant information.

```
samtools faidx mulberry_reference.fasta Super-Scaffold_<ID> | bcftools
consensus <SRRID>_filtered_indels.vcf.gz > Contig_<ID>.fa
```

Expected result

```
>Super-Scaffold_1_31878303
TTACGCAGTTTAATCATTTTCTAAAGTATGAAACTTAGATTCTAATTTCTTATCAGCAAA
AGAAACTAAGATTCTAAAGAAACACACGACCACGTACGCATTTGGTTACCCACAATCTTA
ATTCCAAGAGTTTCAACAGAGTACATTTTACATAAACTACATTATCATGGTTGTATTGTT
ATATGCCAATGTAAATAAGTGTTCACCTTCTCTATCAGGACCAGGAATAATTTCACTTCC
GCACCTAATATCGATAGTGTCCCCATTGTGAATAACAGCCTCGTCAGCAAGGATTGCAGT
TGAACTATACAAATGCAAACAACAATACACCCCACCACAAAAAGTTAAAATTAAAAAATA
TTAAAGAAAAAAGTAAAAGCTTAGGATTATTCTTTAGTAGTGGGGGACCGTTTTGGACCG
TCCGATCTATTATCAAGTTTCGTTAAGTTTCTTTATTTATCTATTTATTTATTTATTTCT
CGATAGGAAATTGGCAATCAATGGTCCAAGAAGTTGGGTGTTTCTTTTATCCTATTACAA
AATCCTCCAAATGCTTATTGTTACTTTCAAACAAAAGCTCATTACTTCACATTGAAACAT
AAAAATTGAAAACCAACAAAGAAAGTTGGTTTACTGAAGAAGTAGCGAACCTCTTATTTT
GCAAAGTAGCAGAACACAAATCCGAATTTTTTTTGATTGTGCACCATGCATGCTCGTCAA
AAGACGACGGCCTGATCACCGAGCATATGACAATCTCATCAGACTTGATCTCAACATCCG
AATGTCTTGAGTCAGATGGAACTGCAACACGGGAAAAGAGAGCCAGAGAAAACTATATTT
ATGCAATTGAAACTGCTAGAAAAAGAACAAGTCTCAGAATTTATAAAAAAATAAATAAAT
AATTCTTTTGTTTTTGTCTGTTTTTAACATGTTTCACATCATTACATAAAGCCTCCATAG
TGCACATCTCTCAATTCTAGCGAGAAGCAAACTATGCACTAGTAGACGAAGGAATTATGA
GGTTGTGTTTGGTAAGACGGAATAAGTTAGGTGATAAAATATGGAGAGGCAAGGGAAAGC
TTTTCTCTAATTGGGTTGACGATGGATTCCGTCTTCCAATCTACCAAACTTTAGAAGGAT
ATTTTAGATCCTTCATCTAATCCCTCCAACCAAACATTTTCATAAGTTTTAAAATTCATC
TACTTACACTCCCACCCCTATAGCTCATTCTAATTTTCCCTCCTACCAAACAAGGCCTTA
AAGGTGAATGTTTGGAGTTTCCAATGGAAGAGGATAAAAGATATCATCTTGATCATGTTT
CTAGGGAACAACAACAACAATCTTTGGATAAAAATAAAAATTAAAAAAAAAAAGCACTTT
TTCTTCTGCCAGACCCATTGAGGAACCAAAGTGTATGCAAAAAATGCAATTTATAGATTA
TATGTAGAACCTGAACACGTTTAAGTAAGCGGGACTAGAAATAAAAATGAAAAATCGAGT
CTTCTGAATCAAAGATATTTGTGGCAGGCAAGTGGGCAATTTGTTGTGACGAACAAGATT
ACCCACAATCCCTTGCTTGGTTCCCAAAATGAGAAAAAAAGATGCAATCTTTCAACAACA
TTGTGCAGTCCGTTAAAAGGGAAAAAAAAAGTAGTCATTTTTTTTTTTTCCATTTATGTT
TGGTAATTGCATTTGAGTTTTGTGGGTTGTATTCATACTGGCAATATTAATACTAACACT
AAAGAGAGAAAGCGAAAAAGAAAGAGAGAGAGAGAGATACCGAGCTTTGCCCAGATTTCT
TGAGGGGATTTTGCGCCCGAGCTTTCTCCGACTTCCGCCATCGCACACAGCAAATTCCCC
TTTTAACGTCCTTGTTCTTATACAATACACAATATATATATATATATATCTGTGTGTGTA
TACAAGGAGAGAGAGAGAGAGAGTTAAGTAGGACAGAGAGACTCTCACAAGTAGGACAGA
GAGACTCTCACAAAAGAATGTTTAGGGCCCGTTCACTTCGCTGATTCGAGTTTTTAATTC
GAGTTGAGATTTGATTTTATGTGAAAGTTTTTTACTGTAGCAAAAAAATTAGATGTGATT
GTTTGTGAGAGCTTTTTACTGTAGCAAAATTCGAATTTTAAACTCGAATAAGCGAAATGA
ACGGAGCCTTAAGTATATCTGTGTGTGTATACAAGGAGAGAGAGAGAGAGAGTTAAGTAG
```