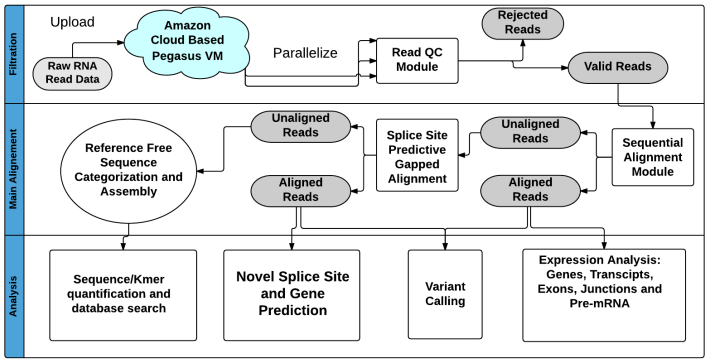
GT-FAR: Users Manual


Filtration
Main Alignement
Analysis
Upload
Raw RNA Read Data
Amazon Cloud Based Pegasus VM
Parallelize
Read QC Module
Rejected Reads
Valid Reads
Sequential Alignment Module
Unaligned Reads
Aligned Reads
Splice Site Predictive Gapped Alignment
Unaligned Reads
Aligned Reads
Reference Free Sequence Categorization and Assembly
Sequence/Kmer quantification and database search
Novel Splice Site and Gene Prediction
Variant Calling
Expression Analysis: Genes, Transcipts, Exons, Junctions and Pre-mRNA

Introduction

Thank you for using GT-FAR. If you are experienced with the shell and don't wish to spend much time reading this manual feel free to quickly compile GT-FAR and type " ./gtfar –help", which will provide a splash screen which gives a very basic overview of the subprograms encapsulated in GT-FAR. For, slightly more documentation, if any of the subprograms are invoked with the flag "–help", more instructions will be given. We hope you find GT-FAR a useful tool.

Table of Contents

Chapter 1

What is GT-FAR: An Overview and Introduction

1.1 What does GT-FAR do?

GT-FAR is a pipeline that analyzes RNA-Seq data and performs (1) genome and transcriptome annotation and file creation for new species, (2) read filtration and quality control, (3) RNA-seq alignment and expression analysis, (4) characterization of novel splice junctions, and (5) genome and transcriptome free analysis of RNA-Seq data. Figure 1 shows a high level description of how GT-FAR may be used.

1.2 How does GT-FAR work?

GT-FAR performs analysis by using many programs.

Chapter 2

Using GT-FAR: From the command line to the cloud

2.1 What is required to run GT-FAR?

GT-FAR can be run using the command line or through a graphical user interface and making use of the amazon cloud or another server. If one doesn't have an amazon cloud account then the requirements to run GTFAR include a unix machine a working verison of python, c++ and bash. Each subprogram in GT-FAR has different requirements. While GT-FAR performs best on a multicore computing cluster; the program is capable of sacrificing speed if memory is not available. At a minimum GT-FAR requires at least 4gb of memory to perform the annotation step (assuming the human genome) and 8gb of memory to perform the analysis and splice generation subprograms. The GTfree analysis requires just 2GB of memory but may perform slow if the file size of read files is very large unless more memory is available.

2.2 Running GT-FAR using a graphical user interface

- will add -

2.3 Running GT-FAR using the command line

Each subprogram of GT-FAR can be called with a different command. Each command has different options which can be invoked. Below we will describe a series of commands that may be used for human data. In the downloaded package there also exists sample data which can be used to run each command quickly to test computational requirements and get a handle on using GT-FAR.

1. **./gtfar easyrun**

   The above command does everything.

2. **./gtfar annotate**

   The above command will is used to index and annotate the genome

3. **./gtfar filter**

   Filter reads.

4. **./gtfar iterMap**

   Mapping and parsing

5. **./gtfar postprocess**

   performs post-processing

6. **./gtfar compare**

   The only subprogram not included in easyrun - this compares multiple outputs

Programs 2-6 are included (when necessary) whenever easyrun is invoked.

Chapter 3

File Descriptions

Chapter 4

Detailed Description of output files

Each output file produced by gtfar is named with a user-specified prefix and a type-dependent suffix.

4.1 Read QC-Ouput

- *$PREFIX*.qc.stats

  The stats file from filtering.

- *$PREFIX*.reject.fastq

  The rejected reads.

4.2 Alignment Output Files

- *$PREFIX*.unmapped.fastq

  The unmapped reads.

- *$PREFIX*.sam

  The iterative alignment and analysis program of gtfar produces a single sam file which contains information about the genomic and transcriptomic location of the reads. This file can further be parsed by custom or supplied analysis scripts to elucidate information about gene or feature expression. Instead of providing an overwhelming amount of data for the user GT-FAR provides a single file which can be viewed with additional viewing software (igv, goldenhelix) or can be quickly parsed and analyzed with custom scripts of supplied scripts. A detailed column by column description of the sam file is shown below:

1. Read ID (The read tag)
2. Positive/Negative Strand (0/16)
3. Chromsome Alignment
4. Chromsome Position
5. Alignment Quality
6. Cigar String
7. *
8. 0
9. 0
10. Read Sequence
11. Quality String
12. NM (Number of mismatches)
13. GT (Genome Type - Score for Unique/Ambiguous Mapping*)
14. TT (Transcriptome Type - Score for Unique/Ambiguous Mapping*)
15. CL (Class
16. Exon, Junction, Intron, Intergenic)

17. GN (Gene Name)
18. AN (Alternate Gene Name)
19. Family (Gene Family
20. Protein Coding, Psuedogene, etc)
21. Strand Identity***
22. RT (Read Tag - user supplied)

- Genome Free File

4.3 Post Alignment Default Sample Processing

4.3.1 Count Files

The post-alignment sample processing module produces multiple count files. All count files follow a similar 3-column format where the three columns describe:

1. Unique Feature Name (eg. Gene Name, Splice Junction Description, etc)
2. Number of sense alignments
3. Number of anti-sense* alignments

*If an unstranded data is used all alignments are considered "sense" and the third column is always zero.

The default count files produced include:

- *$PREFIX*.gene.cnts

  The number of unique alignments per gene model.

  Example Line:

ENSG00000105298.9,CACTIN 5 0

The gene "CACTIN" has five sense alignments

- *$PREFIX*.feature.cnts

  The number of unique alignments in each sub-feature in a gene model (eg. exons, known/novel junctions, introns, etc)

  Example Lines:

  ENSG00000105298.9,CACTIN@KJXN:JXN=3612411,3613056 3 0

  The known junction in CACTIN from 3612411 to 3613056 has three alignments

  ENSG00000105298.9,CACTIN@EXON:COORDS=3610639-3612411 9 0

  The exon spanning 3610639-3612411 has nine alignments

- *$PREFIX*.overlapGene.cnts

  Describes the number of unique alignments which per overlapping gene group.

  Example Line:

  ENSG00000173621.8,LRFN4/ENSG00000173599.9,PC 6 0

  The overlapping genes LRN4 and PC share 6 reads map to a single genomic loci

- *$PREFIX*.ambiguousGene.cnts

  Example Line:

  ENSG00000167615.12,LENG8/ENSG00000182909.5,LENG9 17 0

  The similiar genes LENG8 and LENG9 share 17 read which map to both genomic loci.

4.3.2 Summary Files

GT-FAR produces a single alignment summary file which describes a high level alignment quantification. Each line of the file *$PREFIX*.summary.out describes the number of read alignments which aligned to a certain annotation or chromosome. Some example lines include:

Total-Reads: 500 Unique 400 MultiAnnotated 50 Ambiguous 25

Sense/AntiSense: 450 50

Substitutions 0 300 1 100 2 50 3 30 4 10 5 5 6 5

These lines describe the total number of reads which mapped to unique, overlapping, and ambiguous loci (1), the number of sense/anti-sense alignments (2) and the number of substitutions per alignment (3).