



TELL-Seq™ Data Analysis Software User Guide

**for
Tell-Sort**

Table of Contents

1. Introduction	2
2. Installation	3
3. Run Tell-Sort Pipeline	5
➤ Run Tell-Sort on Linked Read FASTQ Data	5
➤ Prepare Genome VCF Directory (-g parameter)	7

1. Introduction

This document describes instructions on how to use data analysis software Tellysis accompanied with the TELL-Seq WGS Library Prep Kit.

The TELL-Seq WGS library prep kit uses an innovative Transposase Enzyme Linked Long-read Sequencing (TELL-Seq™) technology to prepare a paired-end library to generate barcode linked reads from an Illumina sequencing system. Linked reads can then be processed and analyzed by Tellysis to be used for genome wide variant calling, haplotype phasing, structural variation detection, metagenomic studies and *de novo* sequencing assembly, etc.

Tellysis software comes in the form of three main pipelines.

- **Tell-Read**

a set of pipeline processes that takes as input the sequencing output from an NGS sequencing instrument and generates linked-read FASTQ data, as well as QC reports.

- **Tell-Sort**

a set of pipeline processes that takes as input the linked-read data from Tell-Read result and performs variant calling, phasing and SV.

- **Tell-Link**

de novo assembly pipeline processes that builds barcode-aware assembly graph, assembles contigs and performs scaffolding.

2. Installation

The Tellysis software is currently delivered as Docker images for consistent installations and executions to minimize any potential issues arise from user environment. As such, a Docker running environment is required. For Docker engine installation instructions, user is referred to Docker web site <https://docs.docker.com/install/>.

If you don't already have a Docker running environment, you need to install the Docker engine. Docker is available in two editions: Community Edition (CE) and Enterprise Edition (EE). Following is an example for getting and installing Docker CE for Ubuntu/Debian systems. If you already have a Docker environment, you can skip these steps and go to the next paragraph for installation of Tell-Sort docker image.

Step 1: Update Software Repositories

As usual, it's a good idea to update the local database of software to make sure you've got access to the latest revisions.

Therefore, open a terminal window and type:

```
sudo apt-get update
```

Allow the operation to complete.

Step 2: Uninstall Old Versions of Docker

Next, it's recommended to uninstall any old Docker software before proceeding.

Use the command:

```
sudo apt-get remove docker docker-engine docker.io
```

Step 3: Install Docker

To install Docker on Ubuntu, in the terminal window enter the command:

```
sudo apt install docker.io
```

Step 4: Start and Automate Docker

The Docker service needs to be setup to run at startup. To do so, type in each command followed by enter:

```
sudo systemctl start docker
sudo systemctl enable docker
```

Step 5: Running Docker as a non-root user

If you don't want to preface the `docker` command with `sudo`, create a Unix group called `docker` and add users to it:

```
sudo groupadd docker
sudo usermod -aG docker $USER
```

Step 6: Log out and log back in

After you log back in, you can run Docker as a non-root user.

After the installation of Docker or if you already have a Docker environment, follow the steps below to install Tell-Sort docker image.

1. Download Tell-Sort docker image package `tellsort.tar.gz`.
2. Unzip `tellsort.tar.gz`, this will create a directory `tellsort-release` that contains the docker image of the pipeline named `docker-tellsort`, and a Unix shell script `run_tellsort.sh`.

```
$ tar xzvf tellsort.tar.gz
```

3. Load the docker image

```
$ cd tellsort-release
$ docker load -i docker-tellsort
```

4. Check image `docker-tellsort` is loaded

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker-tellsort	latest	bad2a17180f9	About an hour ago	3.15GB

5. (Optional) To remove the image `docker-tellsort` to upgrade to a newer version

```
$ docker image rm -f bad2a17180f
```

3. Run Tell-Sort Pipeline

Tell-Sort pipeline takes as input from processed fastq data resulted from Tell-Read pipeline (See *TELL-Seq Data Analysis Software User Guide for Tell-Read*).

Tell-Sort pipeline is delivered as a docker image. Tell-Sort package provides wrapper scripts to run the pipeline so users can avoid the docker details.

➤ Run Tell-Sort on Linked Read FASTQ Data

The wrapper script to run Tellsort pipeline is `run_tellsort.sh`. The command line looks like following,

```
$ run_tellsort.sh \  
  -r1 <R1 read>.fastq.gz \  
  -r2 <R2 read>.fastq.gz \  
  -i1 <I1 read>.fastq.gz \  
  -r <genome reference file in fasta> \  
  -o <path/to/output> \  
  -p <prefix name> \  
  -b <genome_variants>.bed \  
  -v <genome_variants>.vcf.gz \  
  -t number of threads
```

-r1	This required parameter specifies read 1 fastq file in gz compressed format.
------------	--

-r2	This optional parameter specifies read 2 fastq file in gz compressed format.
-i1	This required parameter specifies index 1 fastq file in gz compressed format.
-r	This required parameter specifies genome reference file in fasta format.
-o	This required parameter specifies the output directory.
-p	This required parameter specifies a prefix name for identifying a result set.
-g	This required parameter specifies the directory containing genome VCF files of the specie at chromosome level, e.g. “-g /path/to/GRCh38”. See detailed description in “Prepare Genome VCF Directory” below.
-b	This required parameter specifies reference genome variants bed file. This is a standard reference result that the pipeline will compare against with.
-v	This required parameter specifies reference genome variants VCF file. This is a standard reference result that the pipeline will compare against with.
-t	This optional parameter specifies the number of threads. The default is 30.

An example is shown below,

```
$ run_tellsort.sh \
-r1 runTest/Full/runTest_R1_A501.fastq.gz.corrected.fastq.err_barcode_removed.fastq.gz \
-r2 runTest/Full/runTest_R2_A501.fastq.gz.corrected.fastq.err_barcode_removed.fastq.gz \
-i1 runTest/Full/runTest_I1_A501.fastq.gz.corrected.fastq.err_barcode_removed.fastq.gz \
-r /data/genome/DH10b/ecoli_dh10b.fasta \
-o runTestResult \
-p 501 \
-b /home/ubuntu/grch38.protein_coding_genes_variants.bed \
-v /home/ubuntu/GRCh38_GIAB_highconf.NA12878.vcf.gz
```

In this example, the GIAB NA12878 reference VCF files, *grch38.protein_coding_genes_variants.bed*, *GRCh38_GIAB_highconf.NA12878.vcf.gz* can be downloaded from GIAB site, ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/NA12878_HG001/latest/GRCh38/.

The output directory runTestResult will be created and assembly results will be stored in the directory.

```
$ ls -al runTestResult/
drwxrwxr-x 4 ubuntu ubuntu    6144 Jul 21 01:34 ./
drwxrwxr-x 3 ubuntu ubuntu    6144 Jul 19 13:38 ../
```

```
-rw-rw-r-- 1 ubuntu ubuntu 637401457 Jul 20 23:34 501.diploid.vcf
-rw-rw-r-- 1 ubuntu ubuntu 1156487793 Jul 20 23:34 501.vcf
-rw-rw-r-- 1 ubuntu ubuntu 5612 Jul 21 01:33 501.log
drwxrwxr-x 2 ubuntu ubuntu 6144 Jul 21 01:33 501_stats/
drwxrwxr-x 2 ubuntu ubuntu 71680 Jul 21 01:33 501_temp/
```

The summary report `phasing_final_reports.html` is stored in directory `500_stat`.

```
$ ls -al runTestResult/500_stat/
drwxrwxr-x 2 ubuntu ubuntu 6144 Jul 21 01:33 ./
drwxrwxr-x 4 ubuntu ubuntu 6144 Jul 21 01:34 ../
-rw-rw-r-- 1 ubuntu ubuntu 4179 Jul 20 15:33 data.js
-rw-rw-r-- 1 ubuntu ubuntu 649137 Jul 20 19:49 phasing_final_reports.html
-rw-rw-r-- 1 ubuntu ubuntu 3617 Jul 20 19:49 phasing_final_reports.txt
-rw-rw-r-- 1 ubuntu ubuntu 8203 Jul 20 14:04 phasing_results.txt
-rw-rw-r-- 1 ubuntu ubuntu 3653 Jul 20 15:31 plot.html
-rw-rw-r-- 1 ubuntu ubuntu 883 Jul 20 15:33 summary.inf
```

➤ Prepare Genome VCF Directory (`-g` parameter)

Tell-Sort pipeline produces the phasing statistics at the chromosome level for the specie. In order to compare with the reference genome, a VCF reference directory needs to be created to establish the reference VCF at the chromosome level, in addition to the genome level references specified by `-b` and `-v` parameters.

1. Prepare a folder that contains all the reference chromosome VCF files and name it after the genome name. (e.g. GRCh38)
2. Create a text file and name it **chromosomes_{genome name}.txt** and include all the chromosome names in that text file. Put the text file inside of the same folder that contains the chromosome VCF files (e.g. `/path/to/GRCh38/chromosomes_GRCh38.txt`)

```
$ ls -al /home/ubuntu/GRCH38
total 1624188
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 3 16:05 ./
drwxrwxr-x 27 ubuntu ubuntu 4096 Sep 24 13:22 ../
-rw-rw-r-- 1 ubuntu ubuntu 128 Sep 11 06:28 chromosomes_GRCh38.txt
-rw-rw-r-- 1 ubuntu ubuntu 86879423 Sep 11 06:31 GRCh38_chr10.vcf
-rw-rw-r-- 1 ubuntu ubuntu 87526542 Sep 11 06:31 GRCh38_chr11.vcf
-rw-rw-r-- 1 ubuntu ubuntu 77999828 Sep 11 06:31 GRCh38_chr12.vcf
-rw-rw-r-- 1 ubuntu ubuntu 68692074 Sep 11 06:31 GRCh38_chr13.vcf
-rw-rw-r-- 1 ubuntu ubuntu 56595194 Sep 11 06:31 GRCh38_chr14.vcf
-rw-rw-r-- 1 ubuntu ubuntu 48835798 Sep 11 06:31 GRCh38_chr15.vcf
-rw-rw-r-- 1 ubuntu ubuntu 30458753 Sep 11 06:31 GRCh38_chr16.vcf
-rw-rw-r-- 1 ubuntu ubuntu 42534812 Sep 11 06:31 GRCh38_chr17.vcf
```



```
-rw-rw-r-- 1 ubuntu ubuntu 41635063 Sep 11 06:31 GRCh38_chr18.vcf
-rw-rw-r-- 1 ubuntu ubuntu 32849295 Sep 11 06:31 GRCh38_chr19.vcf
-rw-rw-r-- 1 ubuntu ubuntu 135561328 Sep 11 06:31 GRCh38_chr1.vcf
-rw-rw-r-- 1 ubuntu ubuntu 37348542 Sep 11 06:31 GRCh38_chr20.vcf
-rw-rw-r-- 1 ubuntu ubuntu 24049493 Sep 11 06:31 GRCh38_chr21.vcf
-rw-rw-r-- 1 ubuntu ubuntu 19136769 Sep 11 06:31 GRCh38_chr22.vcf
-rw-rw-r-- 1 ubuntu ubuntu 134410383 Sep 11 06:31 GRCh38_chr2.vcf
-rw-rw-r-- 1 ubuntu ubuntu 121443463 Sep 11 06:31 GRCh38_chr3.vcf
-rw-rw-r-- 1 ubuntu ubuntu 107020665 Sep 11 06:31 GRCh38_chr4.vcf
-rw-rw-r-- 1 ubuntu ubuntu 102128061 Sep 11 06:31 GRCh38_chr5.vcf
-rw-rw-r-- 1 ubuntu ubuntu 112250082 Sep 11 06:31 GRCh38_chr6.vcf
-rw-rw-r-- 1 ubuntu ubuntu 94251504 Sep 11 06:31 GRCh38_chr7.vcf
-rw-rw-r-- 1 ubuntu ubuntu 82237554 Sep 11 06:31 GRCh38_chr8.vcf
-rw-rw-r-- 1 ubuntu ubuntu 74125821 Sep 11 06:31 GRCh38_chr9.vcf
-rw-rw-r-- 1 ubuntu ubuntu 45124223 Sep 11 06:31 GRCh38_chrX.vcf
```

```
$ cat /home/ubuntu/GRCH38/chromosomes_GRCh38.txt
```

```
chr1
chr2
chr3
chr4
chr5
chr6
chr7
chr8
chr9
chr10
chr11
chr12
chr13
chr14
chr15
chr16
chr17
chr18
chr19
chr20
chr21
chr22
chrX
```