Quick Manual: Local-Haplotype Variant Calling Software (LocHap -ver1.0)

Subhajit Sengupta[1], Kamalakar Gulukota[2], Yitan Zhu[1], Yuan Ji[1,3,*]

[1] *Center for Biomedical Research Informatics, NorthShore University HealthSystem, Evanston, IL, USA.*

[2] *Center for Molecular Medicine, NorthShore University HealthSystem, Evanston, IL, USA.*

[3] *Department of Health Studies, University of Chicago, Chicago, IL, USA.*

[*] Email for Correspondence: koaeraser@gmail.com

1 Ver-1.0 Release Note

We have tested **LocHap** on mac OSX 10.8.4 and 10.9, CentOs 6.5, and Ubuntu 12.04.4 LTS. Presently we have not tested our software on any Windows operating system.

2 Prerequisites

Installation of **LocHap** requires g++ compiler and GNU Make. If users don't have those already, then for UNIX system they can download them from here: `ftp://ftp.gnu.org/gnu/`. For mac OSX, installing Xcode (IDE for Apple's OSX and iOS) would automatically install those tools.

3 Installation

LocHap is an open-source self-contained computational tool. Standard installation of **LocHap** places the main functions as well as all the required libraries on the user's computer. It also includes a small example for illustration purposes. In the small example, a small *bam* and a *vcf* file are also copied to the user's computer.

> For format specifications of *bam* and *vcf* files, see `http://samtools.github.io/hts-specs/SAMv1.pdf` and `http://samtools.github.io/hts-specs/VCFv4.1.pdf`, respectively.

Installation steps are as follows.

1. Download and extract the "LocHap-release-v1.tar.gz" file from the download section of

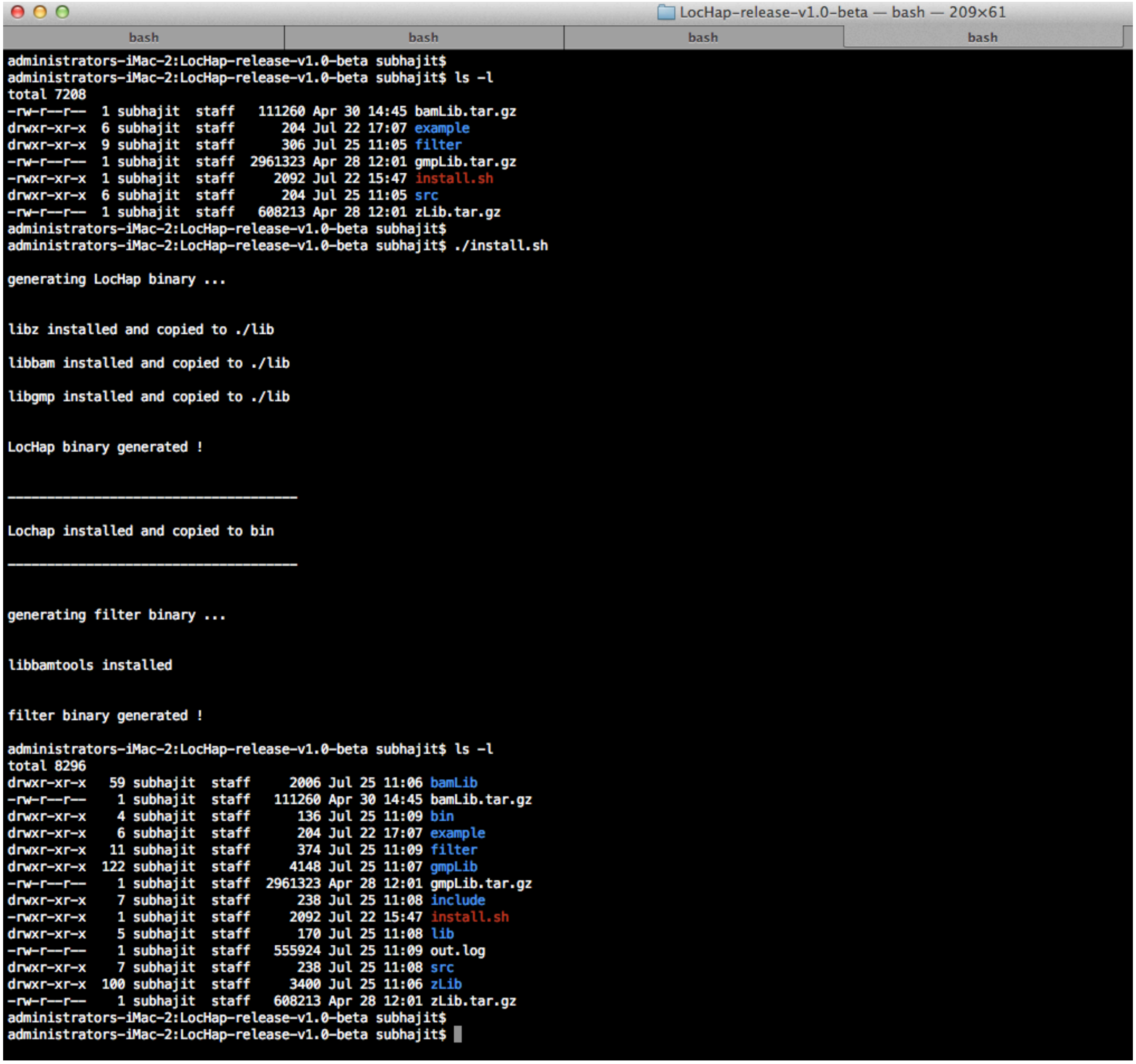
   `http://www.compgenome.org/`

   and by default all the files would be extracted inside the directory named "LocHap-release-v1".

2. In the extracted directory, in a terminal prompt, run the script "install.sh" which is an executable file inside the directory. It would take approximately 5 to 6 minutes to finish the installation process.

3. If successful, **LocHap** binary executable would be generated in the "bin" subdirectory under the directory "LocHap-release-v1".

A screenshot of running the installer is provided below in Fig. 1.



```
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$ ls -l
total 7208
-rw-r--r--  1 subhajit  staff   111260 Apr 30 14:45 bamLib.tar.gz
drwxr-xr-x  6 subhajit  staff      204 Jul 22 17:07 example
drwxr-xr-x  9 subhajit  staff      306 Jul 25 11:05 filter
-rw-r--r--  1 subhajit  staff  2961323 Apr 28 12:01 gmpLib.tar.gz
-rwxr-xr-x  1 subhajit  staff     2092 Jul 22 15:47 install.sh
drwxr-xr-x  6 subhajit  staff      204 Jul 25 11:05 src
-rw-r--r--  1 subhajit  staff   608213 Apr 28 12:01 zLib.tar.gz
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$ ./install.sh

generating LocHap binary ...


libz installed and copied to ./lib

libbam installed and copied to ./lib

libgmp installed and copied to ./lib


LocHap binary generated !


-------------------------------------

Lochap installed and copied to bin

-------------------------------------


generating filter binary ...


libbamtools installed


filter binary generated !

administrators-iMac-2:LocHap-release-v1.0-beta subhajit$ ls -l
total 8296
drwxr-xr-x   59 subhajit  staff     2006 Jul 25 11:06 bamLib
-rw-r--r--    1 subhajit  staff   111260 Apr 30 14:45 bamLib.tar.gz
drwxr-xr-x    4 subhajit  staff      136 Jul 25 11:09 bin
drwxr-xr-x    6 subhajit  staff      204 Jul 22 17:07 example
drwxr-xr-x   11 subhajit  staff      374 Jul 25 11:09 filter
drwxr-xr-x  122 subhajit  staff     4148 Jul 25 11:07 gmpLib
-rw-r--r--    1 subhajit  staff  2961323 Apr 28 12:01 gmpLib.tar.gz
drwxr-xr-x    7 subhajit  staff      238 Jul 25 11:08 include
-rwxr-xr-x    1 subhajit  staff     2092 Jul 22 15:47 install.sh
drwxr-xr-x    5 subhajit  staff      170 Jul 25 11:08 lib
-rw-r--r--    1 subhajit  staff   555924 Jul 25 11:09 out.log
drwxr-xr-x    7 subhajit  staff      238 Jul 25 11:08 src
drwxr-xr-x  100 subhajit  staff     3400 Jul 25 11:06 zLib
-rw-r--r--    1 subhajit  staff   608213 Apr 28 12:01 zLib.tar.gz
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$ 
```

Figure 1: Screenshot of installation process

4 Basic Usage

For general applications of **LocHap**, a user must provide the following two types of input files for biological samples to be analyzed.

- A sorted (coordinate-based) *bam* file and the corresponding bam index file (*bai*).

- A *vcf* file containing SNP information for the sample that the *bam* file characterizes and the one that user wants to run LH analysis.

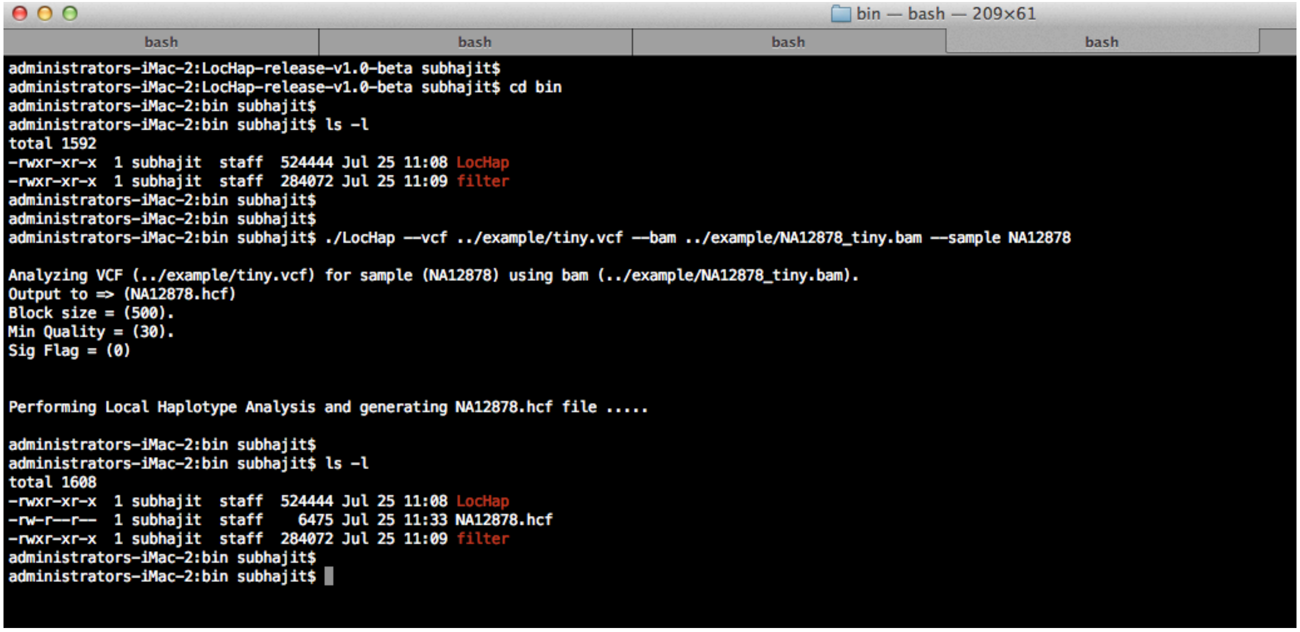
When multiple samples are available for analysis, the *vcf* file must contain all the corresponding *bam* files, and LocHap can be executed in parallel, one per sample.

The main command for executing LocHap for one particular sample is'

```
./LocHap --vcf ⟨full name of the vcf file⟩ --bam ⟨full name of the bam file⟩ --sample ⟨sample name⟩
```

> The ⟨sample name⟩ is a standard field in any vcf file. Users can find the sample name therein. The header line in any *vcf* has 8 fixed, mandatory columns. They are: #CHROM,POS, ID, REF, ALT, QUAL, FILTER, INFO. If genotype data is present in the file, these are followed by a FORMAT column header, then an arbitrary number of sample IDs. The header line is tab-delimited.

Screenshot of this main command is provided below in Fig. 2.



```
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$ cd bin
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ls -l
total 1592
-rwxr-xr-x  1 subhajit  staff  524444 Jul 25 11:08 LocHap
-rwxr-xr-x  1 subhajit  staff  284072 Jul 25 11:09 filter
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ./LocHap --vcf ../example/tiny.vcf --bam ../example/NA12878_tiny.bam --sample NA12878

Analyzing VCF (../example/tiny.vcf) for sample (NA12878) using bam (../example/NA12878_tiny.bam).
Output to => (NA12878.hcf)
Block size = (500).
Min Quality = (30).
Sig Flag = (0)


Performing Local Haplotype Analysis and generating NA12878.hcf file .....

administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ls -l
total 1608
-rwxr-xr-x  1 subhajit  staff  524444 Jul 25 11:08 LocHap
-rw-r--r--  1 subhajit  staff    6475 Jul 25 11:33 NA12878.hcf
-rwxr-xr-x  1 subhajit  staff  284072 Jul 25 11:09 filter
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ 
```

Figure 2: Screenshot for executing the main command of **LocHap**

There are few optional flags that can be added to this main command.

- out: [string] – output file prefix which has default value same as sample name. Otherwise if user provides "output" as the prefix "output.hcf" will be generated.

- sig: [boolean] – significant flag which sets a flag for printing only DNA segments (see main paper for definition of a DNA "segment") with at least one significant haplotype called. The default value is currently set to 0, so all segments will be printed.

- size: [integer] – Max base pairs between two adjacent SNVs. This is an important value and is denoted as K in the paper. See §2.1 in Online Methods (default is 500 NTs and must be between 50 and 1000).

- qual: [integer] LocHap ignores all reads with Phred-scaled Mapping Quality less than qual. LocHap Ignores all bases with Base Quality less than qual (default is set to 30 and it must be greater than 15)

- igv: [boolean] – IGV compatibility flag which sets a flag for printing in format suitable for loading into IGV software from Broad Institute. By default the output is written in native *hcf* format and the output file name would be ⟨sample name⟩.hcf or ⟨sample name⟩.igv, based on the choice of the flag.

LocHap handles one bam at a time and that sample has to be inside the *vcf*; it does not care if other samples are inside that *vcf.* For analyzing multiple samples, one can parallelize the analyses, one for each sample.

4.1 Demo Example

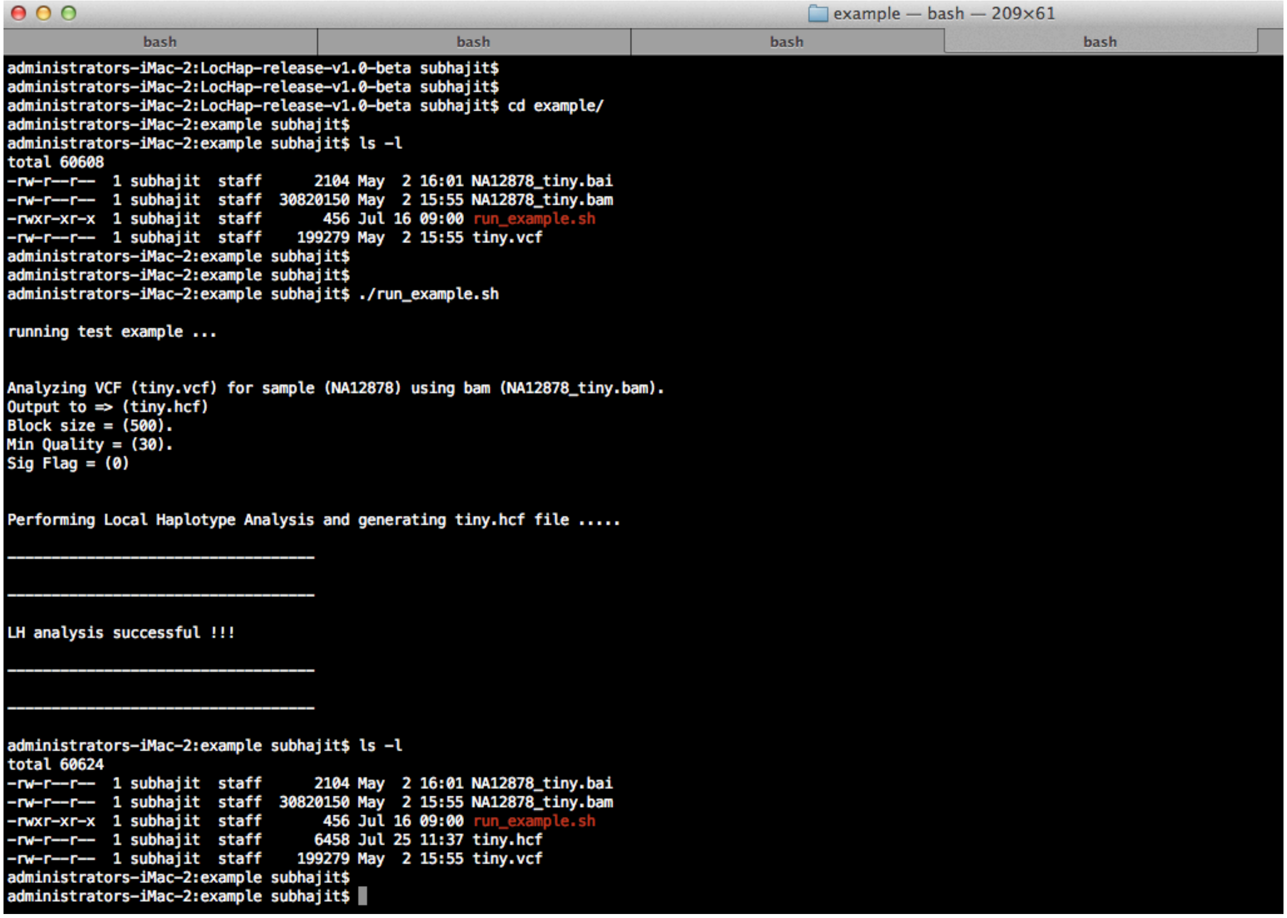
In the "Example" subdirectory under the directory "LocHap-release-v1", a test example is prepared to demonstrate the main functions of **LocHap**, which can be executed by the following steps:

- Go to the "Example" subdirectory.

- Execute the script "run-example.sh" (Linux command "./run-example.sh").

- If successful, an output *hcf* file will be produced (called "tiny.hcf") with a message on the screen (see Fig. 3). The "tiny.hcf" file is an ASCII tab-delimited text file (see Online Methods for detail) containing all the local haplotype variants produced from the proposed statistical inference. See section 5 for detail of a *hcf* file.

A screenshot is provided below in Fig. 3 for the commands above.



```
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$ cd example/
administrators-iMac-2:example subhajit$
administrators-iMac-2:example subhajit$ ls -l
total 60608
-rw-r--r--  1 subhajit  staff      2104 May  2 16:01 NA12878_tiny.bai
-rw-r--r--  1 subhajit  staff  30820150 May  2 15:55 NA12878_tiny.bam
-rwxr-xr-x  1 subhajit  staff       456 Jul 16 09:00 run_example.sh
-rw-r--r--  1 subhajit  staff    199279 May  2 15:55 tiny.vcf
administrators-iMac-2:example subhajit$
administrators-iMac-2:example subhajit$
administrators-iMac-2:example subhajit$ ./run_example.sh

running test example ...


Analyzing VCF (tiny.vcf) for sample (NA12878) using bam (NA12878_tiny.bam).
Output to => (tiny.hcf)
Block size = (500).
Min Quality = (30).
Sig Flag = (0)


Performing Local Haplotype Analysis and generating tiny.hcf file .....

-----------------------------------

-----------------------------------


LH analysis successful !!!

-----------------------------------

-----------------------------------


administrators-iMac-2:example subhajit$ ls -l
total 60624
-rw-r--r--  1 subhajit  staff      2104 May  2 16:01 NA12878_tiny.bai
-rw-r--r--  1 subhajit  staff  30820150 May  2 15:55 NA12878_tiny.bam
-rwxr-xr-x  1 subhajit  staff       456 Jul 16 09:00 run_example.sh
-rw-r--r--  1 subhajit  staff      6458 Jul 25 11:37 tiny.hcf
-rw-r--r--  1 subhajit  staff    199279 May  2 15:55 tiny.vcf
administrators-iMac-2:example subhajit$
administrators-iMac-2:example subhajit$ 
```

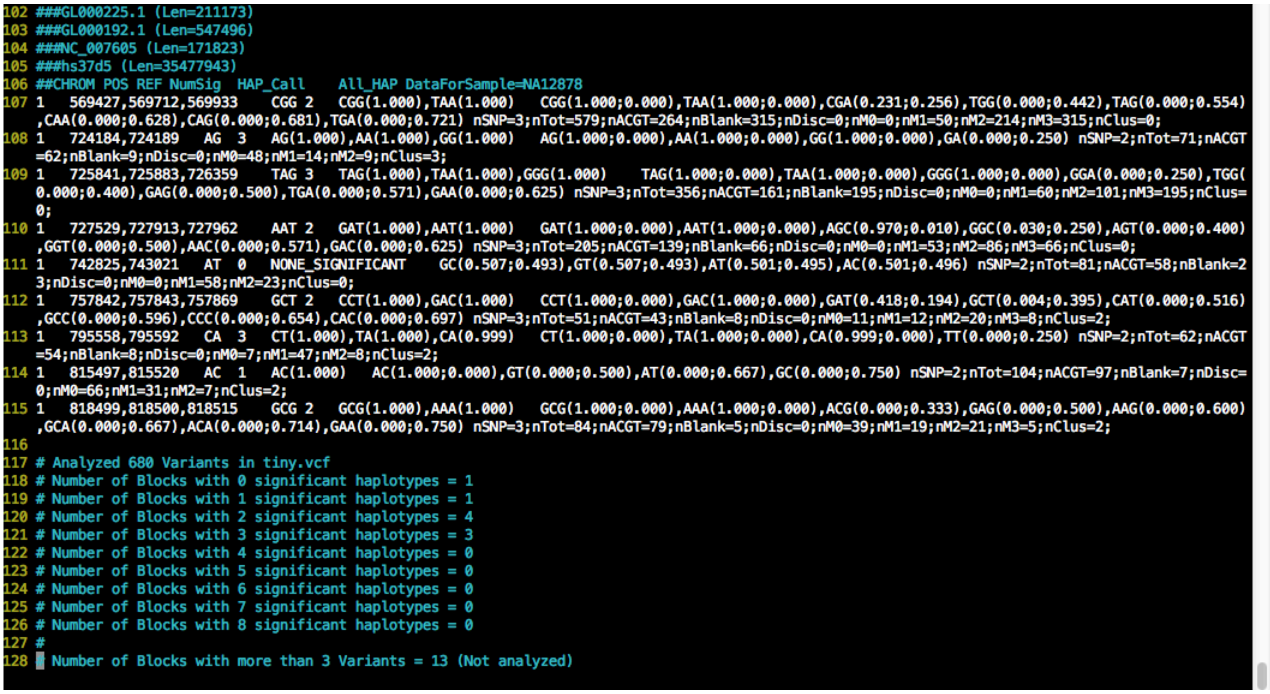
Figure 3: Screenshot for testing an example

5 Haplotype Call Format (*hcf*)

The output segments from **LocHap** are written in the Haplotype Call Format (*hcf*). See Figure 4. Similar to *vcf*, an *hcf* file is a tab-delimited text file. There are a few header fields that contain the field names and their descriptions. After the initial header fields, each line in the *hcf* file represents a local haplotype (might not be a variant) and has seven column fields, displayed from left to right as: chromosome name (CHROM), positions on the chromosome (POS), nucleotides at those positions in the reference genome (REF), number of significant haplotypes (NumSig), called haplotypes (HAP-Call), all the possible haplotypes (All-HAP), data for the sample (DataForSample=⟨sample-name⟩).

Most of the fields are self-explanatory except the following – in the "Hap-Call" field, we include the posterior probability of each haplotype variant that is deemed statistically significant. The "All-HAP" field contains the posterior probability (before semicolon) and corresponding posterior false discovery rate (FDR) (after semicolon) for each possible haplotype. Haplotype variants in the "Hap-Call" field are generated from those in “All-HAP” by using an FDR threshold of 0.01.

In the last field, a few basic statistics about the input-data are given, which include total number of SNPs (nSNP), total number of reads (nTot), number of reads having at least one entry in one of the SNP position (nACGT), number of blank reads (nBlank), number of discrepant reads (nDisc), number of reads with no missing entries, missing entries in one position or two positions or three positions (nM0, nM1, nM2, nM3) and number of clusters directly observed from the data (nClus). Note that, number of unique groups of data that has no missing SNP defines the number of clusters.



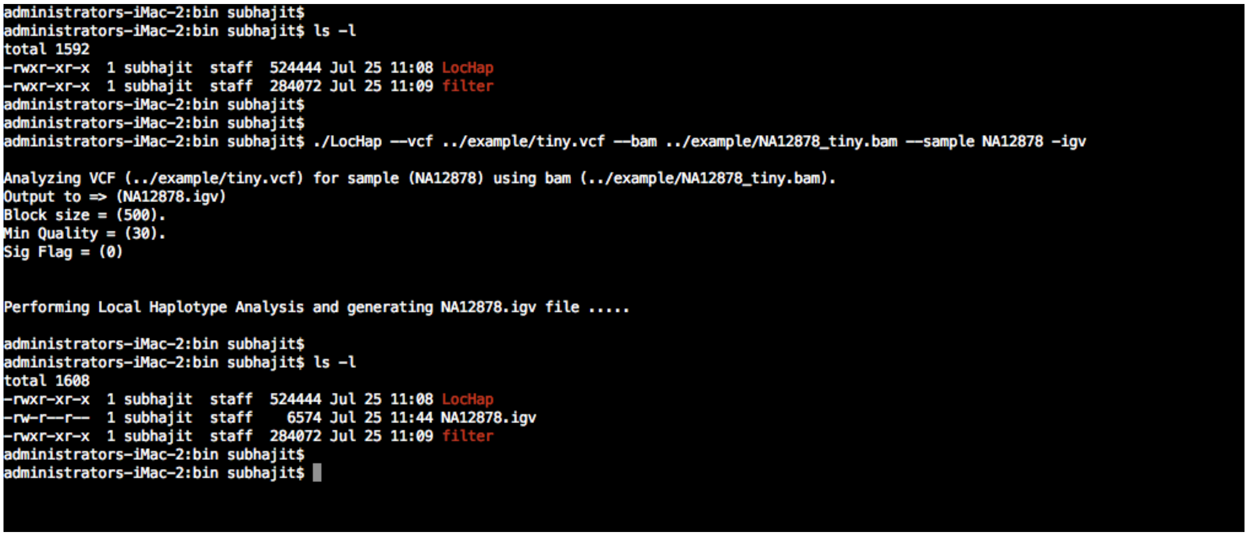
```
###GL000225.1 (Len=211173)
###GL000192.1 (Len=547496)
###NC_007605 (Len=171823)
###hs37d5 (Len=35477943)
##CHROM POS REF NumSig  HAP_Call    All_HAP DataForSample=NA12878
1   569427,569712,569933    CGG 2   CGG(1.000),TAA(1.000)   CGG(1.000;0.000),TAA(1.000;0.000),CGA(0.231;0.256),TGG(0.000;0.442),TAG(0.000;0.554)
,CAA(0.000;0.628),CAG(0.000;0.681),TGA(0.000;0.721) nSNP=3;nTot=579;nACGT=264;nBlank=315;nDisc=0;nM0=0;nM1=50;nM2=214;nM3=315;nClus=0;
1   724184,724189   AG  3   AG(1.000),AA(1.000),GG(1.000)   AG(1.000;0.000),AA(1.000;0.000),GG(1.000;0.000),GA(0.000;0.250) nSNP=2;nTot=71;nACGT
=62;nBlank=9;nDisc=0;nM0=48;nM1=14;nM2=9;nClus=3;
1   725841,725883,726359    TAG 3   TAG(1.000),TAA(1.000),GGG(1.000)    TAG(1.000;0.000),TAA(1.000;0.000),GGG(1.000;0.000),GGA(0.000;0.250),TGG(
0.000;0.400),GAG(0.000;0.500),TGA(0.000;0.571),GAA(0.000;0.625) nSNP=3;nTot=356;nACGT=161;nBlank=195;nDisc=0;nM0=0;nM1=60;nM2=101;nM3=195;nClus=
0;
1   727529,727913,727962    AAT 2   GAT(1.000),AAT(1.000)   GAT(1.000;0.000),AAT(1.000;0.000),AGC(0.970;0.010),GGC(0.030;0.250),AGT(0.000;0.400)
,GGT(0.000;0.500),AAC(0.000;0.571),GAC(0.000;0.625) nSNP=3;nTot=205;nACGT=139;nBlank=66;nDisc=0;nM0=0;nM1=53;nM2=86;nM3=66;nClus=0;
1   742825,743021   AT  0   NONE_SIGNIFICANT    GC(0.507;0.493),GT(0.507;0.493),AT(0.501;0.495),AC(0.501;0.496) nSNP=2;nTot=81;nACGT=58;nBlank=2
3;nDisc=0;nM0=0;nM1=58;nM2=23;nClus=0;
1   757842,757843,757869    GCT 2   CCT(1.000),GAC(1.000)   CCT(1.000;0.000),GAC(1.000;0.000),GAT(0.418;0.194),GCT(0.004;0.395),CAT(0.000;0.516)
,GCC(0.000;0.596),CCC(0.000;0.654),CAC(0.000;0.697) nSNP=3;nTot=51;nACGT=43;nBlank=8;nDisc=0;nM0=11;nM1=12;nM2=20;nM3=8;nClus=2;
1   795558,795592   CA  3   CT(1.000),TA(1.000),CA(0.999)   CT(1.000;0.000),TA(1.000;0.000),CA(0.999;0.000),TT(0.000;0.250) nSNP=2;nTot=62;nACGT
=54;nBlank=8;nDisc=0;nM0=7;nM1=47;nM2=8;nClus=2;
1   815497,815520   AC  1   AC(1.000)   AC(1.000;0.000),GT(0.000;0.500),AT(0.000;0.667),GC(0.000;0.750) nSNP=2;nTot=104;nACGT=97;nBlank=7;nDisc=
0;nM0=66;nM1=31;nM2=7;nClus=2;
1   818499,818500,818515    GCG 2   GCG(1.000),AAA(1.000)   GCG(1.000;0.000),AAA(1.000;0.000),ACG(0.000;0.333),GAG(0.000;0.500),AAG(0.000;0.600)
,GCA(0.000;0.667),ACA(0.000;0.714),GAA(0.000;0.750) nSNP=3;nTot=84;nACGT=79;nBlank=5;nDisc=0;nM0=39;nM1=19;nM2=21;nM3=5;nClus=2;

# Analyzed 680 Variants in tiny.vcf
# Number of Blocks with 0 significant haplotypes = 1
# Number of Blocks with 1 significant haplotypes = 1
# Number of Blocks with 2 significant haplotypes = 4
# Number of Blocks with 3 significant haplotypes = 3
# Number of Blocks with 4 significant haplotypes = 0
# Number of Blocks with 5 significant haplotypes = 0
# Number of Blocks with 6 significant haplotypes = 0
# Number of Blocks with 7 significant haplotypes = 0
# Number of Blocks with 8 significant haplotypes = 0
#
# Number of Blocks with more than 3 Variants = 13 (Not analyzed)
```

Figure 4: Screenshot of a sample *hcf* file.

6 Integrated Genome Viewer (IGV) Compatibility

We have an optional “igv” flag to generate an output file (with extension .igv). The “.igv” output files can be visualized using IGV. When loaded, SNPs belonging to an LHV are shown in *red* bars while those not in a LHV are shown in *blue* bars. Screenshot of this command is provided below in Fig. 5.



```
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ls -l
total 1592
-rwxr-xr-x  1 subhajit  staff  524444 Jul 25 11:08 LocHap
-rwxr-xr-x  1 subhajit  staff  284072 Jul 25 11:09 filter
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ./LocHap --vcf ../example/tiny.vcf --bam ../example/NA12878_tiny.bam --sample NA12878 -igv

Analyzing VCF (../example/tiny.vcf) for sample (NA12878) using bam (../example/NA12878_tiny.bam).
Output to => (NA12878.igv)
Block size = (500).
Min Quality = (30).
Sig Flag = (0)


Performing Local Haplotype Analysis and generating NA12878.igv file .....

administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ls -l
total 1608
-rwxr-xr-x  1 subhajit  staff  524444 Jul 25 11:08 LocHap
-rw-r--r--  1 subhajit  staff    6574 Jul 25 11:44 NA12878.igv
-rwxr-xr-x  1 subhajit  staff  284072 Jul 25 11:09 filter
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$
```

Figure 5: Screenshot for executing the command in order to generate IGV compatible format

When multiple samples are analyzed by **LocHap**, for each sample an *hcf* file will be produced. IGV can open multiple *hcf* files as shown in Fig 6.

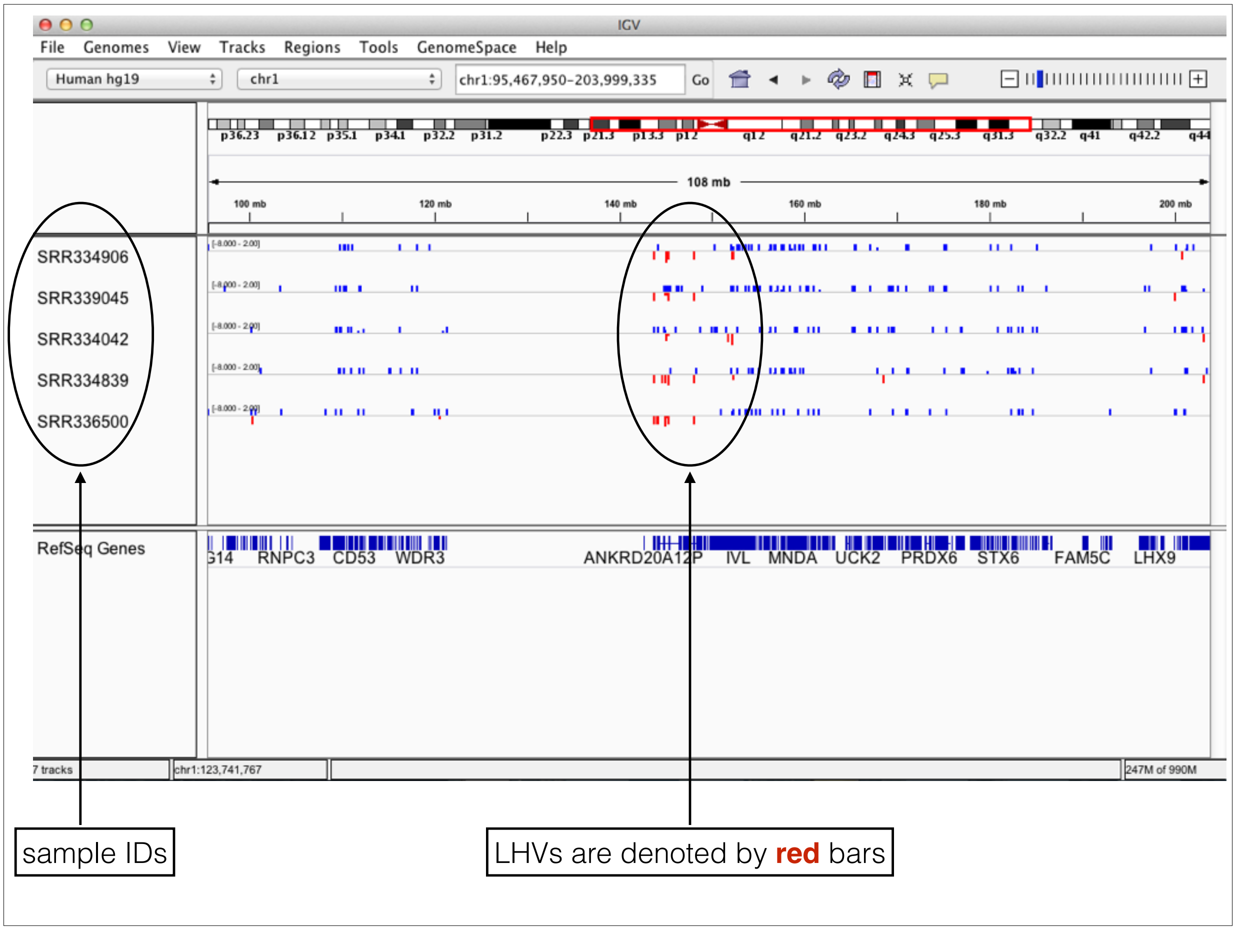

IGV
File Genomes View Tracks Regions Tools GenomeSpace Help
Human hg19
chr1
chr1:95,467,950-203,999,335
Go
p36.23 p36.12 p35.1 p34.1 p32.2 p31.2 p22.3 p21.3 p13.3 p12 q12 q21.2 q23.2 q24.3 q25.3 q31.3 q32.2 q41 q42.2 q44
108 mb
100 mb
120 mb
140 mb
160 mb
180 mb
200 mb
SRR334906
SRR339045
SRR334042
SRR334839
SRR336500
RefSeq Genes
514 RNPC3 CD53 WDR3 ANKRD20A12P IVL MNDA UCK2 PRDX6 STX6 FAM5C LHX9
7 tracks
chr1:123,741,767
247M of 990M
sample IDs
LHVs are denoted by red bars

Figure 6: Screenshot of outputs from multiple samples opened in the IGV viewer

7 Optional Filtering

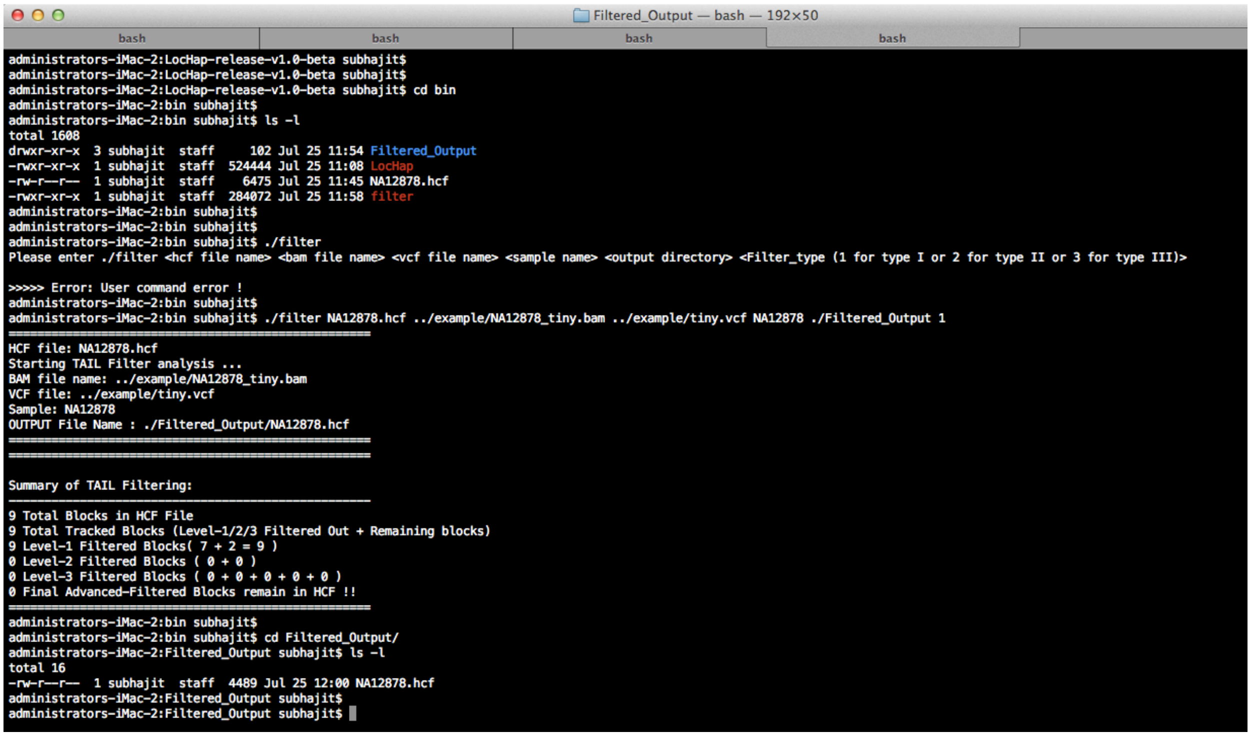
In order to perform type I or type II or type III filtering as mentioned in §6.1 of our Online Methods, a binary executable file named **filter** is also generated at the time of the installation inside the "bin" subdirectory under the directory "LocHap-release-v1". A user needs to provide the following files (in the strict order required by the program) in order to carry out this optional filtering:

- An *hcf* file generated by **LocHap** for one particular sample.
- A sorted (coordinate-based) *bam* file and the corresponding bam index file (*bai*) for that sample.
- A *vcf* file containing SNP information for that sample.

The main command for executing the filter for one particular sample is:

```
./filter ⟨full name of the hcf file⟩ ⟨full name of the bam file⟩ ⟨full name of the vcf
file⟩ ⟨sample name⟩ ⟨full name of output directory⟩ ⟨filter type⟩
```

User can run the filter according to the choice of the stringency level for the filtering. He or she can enter 1 or 2 or 3 in ⟨filter type⟩ in order to run type I or type II or type III filter, respectively.
In Figure 7, screenshot of the command for executing type I filter. For type II or type III filter, the last field of the command needs to be replaced by 2 or 3, respectively while other fields remain exactly the same.



```
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$
administrators-iMac-2:LocHap-release-v1.0-beta subhajit$ cd bin
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ls -l
total 1608
drwxr-xr-x  3 subhajit  staff     102 Jul 25 11:54 Filtered_Output
-rwxr-xr-x  1 subhajit  staff  524444 Jul 25 11:08 LocHap
-rw-r--r--  1 subhajit  staff    6475 Jul 25 11:45 NA12878.hcf
-rwxr-xr-x  1 subhajit  staff  284072 Jul 25 11:58 filter
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ./filter
Please enter ./filter <hcf file name> <bam file name> <vcf file name> <sample name> <output directory> <Filter_type (1 for type I or 2 for type II or 3 for type III)>

>>>>> Error: User command error !
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ ./filter NA12878.hcf ../example/NA12878_tiny.bam ../example/tiny.vcf NA12878 ./Filtered_Output 1
===================================================
HCF file: NA12878.hcf
Starting TAIL Filter analysis ...
BAM file name: ../example/NA12878_tiny.bam
VCF file: ../example/tiny.vcf
Sample: NA12878
OUTPUT File Name : ./Filtered_Output/NA12878.hcf
===================================================
===================================================

Summary of TAIL Filtering:
---------------------------------------------------
9 Total Blocks in HCF File
9 Total Tracked Blocks (Level-1/2/3 Filtered Out + Remaining blocks)
9 Level-1 Filtered Blocks( 7 + 2 = 9 )
0 Level-2 Filtered Blocks ( 0 + 0 )
0 Level-3 Filtered Blocks ( 0 + 0 + 0 + 0 + 0 )
0 Final Advanced-Filtered Blocks remain in HCF !!
===================================================
administrators-iMac-2:bin subhajit$
administrators-iMac-2:bin subhajit$ cd Filtered_Output/
administrators-iMac-2:Filtered_Output subhajit$ ls -l
total 16
-rw-r--r--  1 subhajit  staff  4489 Jul 25 12:00 NA12878.hcf
administrators-iMac-2:Filtered_Output subhajit$
administrators-iMac-2:Filtered_Output subhajit$
```

Figure 7: Screenshot of executing the command for optional filtering

Note that, in the provided example, type I filter would remove all the segments from the *hcf* file that is generated by **LocHap**. The output file generated after filtering is in the same format as the original *hcf* file.

8 Tips

- Any *bam* file should be indexed and the index file (*.bai*) should be in the same directory with the original *bam* file.
    - In case there is no index file, it can be easily generated by executing the command below:

```
samtools index ⟨name of the bam file⟩
```

See `http://samtools.sourceforge.net/samtools.shtml` for command details.

- An input *vcf* file may consist of variant calls for multiple samples. To analyze all the samples, run **LocHap** in parallel using multiple nodes, one per sample.