ExIBD v1.0

Wenqing Fu
Department of Genome Sciences
University of Washington

Dec 8, 2016

1. Introduction

ExIBD is a software package for detecting Identity-by-descent (IBD) segments in exome sequencing data. ExIBD can analyze large-scale datasets with thousands of individuals. ExIBD detects IBD segments in exome sequencing data by three steps. The first step (identification) performs a genome-wide scan using Beagle fastIBD for candidate IBD segments. The second step (refinement) uses Beagle IBD to refine the breakpoint of the candidate IBD segments identified in step 1. The third step (filtering) controls the proportion of reported IBD segments that are false positives. Specifically, IBD segments are filtered out if it spans a genomic region where the corresponding locus specific false discovery rate (FDR) exceeds a desired cutoff.

If you use ExIBD and publish your analysis, please cite the following publication:

Fu, W., Browning, S.R., Browning, B.L., and Akey, J.M. (2016) Robust Inference of Identity by Descent from Exome Sequencing Data. ***Am. J. Hum. Genet.*** 99: 1106-1116

2. Download and Installment

The software package ExIBD is open-source and freely available and can be downloaded from http://akeylab.gs.washington.edu/downloads.html

The software package ExIBD consists of several parts. The **src** directory contains all of the source codes for building the package. The **bin_Linux64** directory contains the precompiled executable files for Linux (64-bit). The **example** directory contains an example with the input files for genotypes, markers, and samples. The **reference** directory contains a reference file called 'FDRref.txt', which is required for FDR filtering. The core of ExIBD is based on BEAGLE v3.3.2 written by Brian L. Browning. Its executable file and manual can be found in the **beagle_v3.3.2** directory. More details of BEAGLE v3.3.2 can be learnt from the BEAGLE web site: https://faculty.washington.edu/browning/beagle/b3.html. Note that the higher version of BEALGLE is not compatible with ExIBD. The **utilities** directory contains a tool calRec, which can be used to convert physical position (GRCh37, hg19) to genetic position.

ExIBD can run step by step or as a pipeline on most UNIX-like operating systems to detect IBD segments in exome sequencing data. If you'd like to compile the ExIBD executable files by yourself, you can do as follows.

1) Make sure the following softwares are available before running or compiling ExIBD

a. A Java 1.6 interpreter (or a later version)

b. A g++ compiler with the support of OpenMP to allow parallel computing

c. Python 2.7 or higher with the package *pandas*

d. If you'd like to run ExIBD as a pipeline, please install PyInstaller (http://www.pyinstaller.org) that can make the python script executable.

2) Compile

a. If you'd like to run ExIBD step by step, compile the programs by typing:

```
$ make Conly
$ make clean_Conly
```

b. If you'd like to run ExIBD as a pipeline, compile the programs by typing:

```
$ make
$ make clean
```

Generally, the first step (identification) requires more memory, and the second step (refinement) requires more CPU cores. Thus, it is more computationally efficient to run ExIBD step by step.

3. Input Files

Three input files are required for the users, i.e., genotypes file, markers file, and samples file, separate files for each chromosome. ExIBD also requires a reference file ('FDRref.txt') that lists the locus specific FDR to detect IBD segments in exome sequencing data for three major continental groups (i.e., African, European, and East Asian populations), which can be found in the reference directory.

3.1 Name rules

The name for genotypes file is arbitrary, such as abc.chr1.gg. But the names for the markers file and the samples file should be same to the full name of the genotypes file with the suffix of '.info' and '.rpop', respectively, such as abc.chr1.gg.info for the markers file and abc.chr1.gg.rpop for the samples file.

3.2 Genotypes file format

Theoretically, all kinds of genotype file supported by BEAGLE v3.3.2 can be used in ExIBD. However, only unphased unrelated data has been tested. Here is an example of a Beagle genotypes file with three individuals and three genotyped markers:

Example 1 – Genotypes file

I	Id	E0001	E0001	H0001	H0001	A0001	A0001
M	1234675	A	G	G	G	A	G
M	1234686	T	T	T	C	T	T
M	1235760	G	T	T	T	G	T

In the example above, rows are variables and columns are individuals. The first column describes the data on each row, for example, an "I" denotes sample identifier ID, and an "M" denotes marker data. The second column contains the name of the variable whose data is given on each row. Variable names should be unique. Unlike the Beagle format, ExIBD uses physical position of the marker as the Marker identifier. In Example 1 there are two columns for each individual: columns 3-4 give data for the first individual, columns 5-6 give data for the second individual, and so on.

In Example 1, the first row (I….) is called a sample identifier line and gives an identifier for each column of data. A sample identifier row is required for ExIBD. The last three rows are marker rows that give marker alleles for the three markers. Note that an identifier should be given for each allele (column). For diploid data, the identifier will typically be the same for both alleles. More details about Beagle genotypes file can be learnt from the Beagle manual, *Beagle_3.3.2_31Oct11.pdf*.

3.3 Markers file format

The name for the markers file should be same to the full name of the genotypes file with the suffix of '.info'. A markers file is required by ExIBD, and one file for each chromosome.

Each row of the markers file will contain four white-space delimited fields. The first field is the marker's physical position, which is also treated as the marker identifier. The second field is the marker genetic position in centiMorgan scale. The remaining two fields are the marker alleles. The markers must be given in chromosomal order. The markers file must contain as many as markers as are in the genotypes file.

Example 2 – Markers file

```
1234675   4.2749   A   G
1234686   4.2749   C   T
1235760   4.2755   G   T
```

3.4 Samples file format

The name for the samples file should be same to the full name of the genotypes file with the suffix of '.rpop'. A samples file is required by ExIBD, and one sample for each row.

Each row of the samples file will contain two white-space delimited fields. The first field is the sample identifier. The second field indicates the reference population for the sample, whose genetic distance is the closest to the sample's ancestry. In ExIBD v1.0, the reference population can be selected from three major continental groups (i.e., African, European, and East Asian populations), referred to as AFR, EUR and EAS in the samples file.

Example 3 – Samples file

```
E0001   EUR
H0001   EAS
A0001   AFR
```

3.5 Examples

An example with the input files for genotypes (i.e., example.chr19.gt), markers (i.e., example.chr19.gt.info), and samples (i.e., example.chr19.gt.rpop) can be found in ./example. It includes 500 bi-allelic SNVs in chromosome 19 identified in 6,515 exomes. The missing data is annotated as 0.

4. Getting Started: Running Step by Step

Before running ExIBD step by step, please copy beagle.jar, ExIBD_Candidate (or ExIBD_Candidated.py), ExIBD_Refined, ExIBD_Filtered (or ExIBD_Filtered.py), and FDRref.txt into the same directory where the input files exist.

4.1 The first step (identification)

4.1.1 Beagle fastIBD

The genome-wide scan of candidate IBD segments in exome sequencing data can directly performed by Beagle fastIBD. The implementation of Beagle fastIBD can be learnt from the Beagle manual. Multiple independent runs are suggested. Different random seed should be set for each run by the argument seed=<random seed> in Beagle. The prefix for the output filename of Beagle fastIBD is required to be continuous integer and starts from 0. It can be set by the argument out=<output file prefix> like out=0, out=1, …

4.1.2 ExIBD_Candidate

After running Beagle fastIBD, a python script called 'ExIBD_Candidate.py' or its executable file ExIBD_Candidate can be used to combine results from multiple fastIBD runs. If there are overlaps among IBD segments shared by the same individual pair from multiple runs, only one IBD segment with the lowest fastIBD score is reported by ExIBD_Candidate.

Inputs for ExIBD_Candidate:

The original fastIBD output files, such as 0.<the full name of the genotypes file>.fibd.gz, 1.<the full name of the genotypes file>.fibd.gz, etc, are required.

Command line for ExIBD_Candidate:

$ python ./ExIBD_Candidate.py --file <filename> --round <number of fastIBD runs> --fastibdthreshold <score threshold>

or

$ ./ExIBD_Candidate --file <filename> --round <number of fastIBD runs> --fastibdthreshold <score threshold>

Arguments for ExIBD_Candidate:

-h, --help: print this help and exit

-f, --file: the full name of the genotypes file, required

-r, --round: the number of Beagle fastIBD runs (optional, default: 10)

-t, --fastibdthreshold: fastIBD score threshold that controls fastIBD output (optional, default: 1e-10)

Outputs for ExIBD_Candidate:

Two files (with the suffix of '.fibd' and '.candidate', respectively) are produced after the implementation of ExIBD_Candidate. Both of them report the identified IBD segments shared by pairs of samples within the corresponding input file that have fastIBD score less than the threshold specified by the *fastibdthreshold* parameter, after the combination of multiple Beagle fastIBD runs.

The fastIBD output file, named <the full name of the genotypes file>.fibd, has five columns. The first two columns list the two sample identifier for the shared IBD segment. The next two columns list the starting (inclusive) and ending (inclusive) marker indices for the shared IBD segment. The first marker has index 0. The last column gives the fastIBD score for the shared IBD segment.

The candidate IBD file, named <the full name of the genotypes file>.candidate, has four fields. The first two columns list the starting (inclusive) and ending (inclusive) marker indices for the shared IBD segment(s). The third column lists the number of individual pairs who shared the IBD segment in the corresponding region. The fourth field delimited by white-space lists the sample identifiers for these individual pairs. This file is one of the input files for the next step (refinement).

4.2 The second step (refinement)

ExIBD_Refined can be used to refine the endpoints of the candidate IBD segments by calling Beagle IBD. Multiple independent runs are suggested. The highest IBD probability for each marker from multiple runs is kept for the calling of IBD segments. An IBD segment is called when the IBD probability for any position at the corresponding segment exceeds a threshold specified by the *cutoff* parameter.

Inputs for ExIBD_Refined:

The genotypes file, the markers file, and the candidate IBD file are required.

Command line for ExIBD_Refined:

$ ./ExIBD_Refined --file <filename> --numMarker <number of markers> --numFIBD <number of candidate IBD regions> –beagle <basic commands for Beagle> [OPTIONS]

Arguments for ExIBD_Refined:

-h, --help: print this help and exit

-f, --file: the full name of the genotypes file, required

-l, --numMarker: the number of markers, required

-n, --numFIBD: the number of rows for the candidate IBD file, required

-b, --beagle: the basic command for Beagle, where the path of java, the memory requirement, and some arguments for Beagle (such as unphased, phased, trios, pairs, like, missing, maxlr, niterations, nsamples, gprobs, lowmem, excludecolumns, nimputations, redundant) can be set here. **Note that double quotation marks are**

required for this command, for example, --beagle "java –Xmx2048m –jar beagle.jar unphased=example.chr19.gt missing=0"; required

-r, --round: the number of Beagle IBD runs (optional, default: 5)

-e, --extend: the number of markers extended from the breakpoints of candidate IBD segments during the refinement process (optional, default: 0)

-u, --ibd2nonibd: the transition rate from IBD to non-IBD per cM for each sample (optional, default: 0.01)

-v, --nonibd2ibd: the transition rate from non-IBD to IBD per cM for each sample (optional, default: 0.0001)

-w, --ibderror: the estimated genotype error rate when estimating IBD (optional, default: 0.005)

-s, --ibdscale: a tuning parameter that controls the complexity of the haplotype frequency model when performing IBD analysis (optional, default: 2.0)

-p, --cutoff: the cutoff of IBD probability for each position during the calling of IBD segments (optional, default: 0.5)

-t, --threads: the number of threads of execution (optional, default: 1)

Outputs for ExIBD_Refined:

An refined IBD output file with the suffix '.ribd' is produced after the implementation of ExIBD_Refined. It reports the refined IBD segments shared by pairs of samples within the corresponding input file that the IBD probability at any position exceeds the threshold specified by the *cutoff* parameter, after the combination of multiple Beagle IBD runs. This file has four columns. The first two columns list the two sample identifier for the shared IBD segment. The next two columns list the starting (inclusive) and ending (inclusive) marker indices for the shared IBD segment. This file is one of the input files for the next step (filtering).

4.3 The third step (filtering)

A python script called 'ExIBD_Filtered.py' or its executable file ExIBD_Filtered can be used to filter out the refined IBD segment if it spans a genomic region where the corresponding locus specific FDR exceeds a desired cutoff.

Inputs for ExIBD_Filtered:

The refined IBD file, the markers file, the samples file and the reference file are required.

Command line for ExIBD_Filtered:

$ python ./ExIBD_Filtered.py --file <filename> --chr <No. chromosome> --reference <the reference file> --fdr <cutoff of FDR>

or

$ ./ExIBD_Filtered --file <filename> --chr <No. chromosome> --reference <the reference file> --fdr <cutoff of FDR>

Arguments for ExIBD_Filtered:

-h, --help: print this help and exit

-f, --file: the full name of the genotypes file, required

-c, --chr: the chromosome name, required

-r, --reference: the reference file for the pre-assigned locus specific FDRs. Note that the default reference file is based on 10 independent runs of Beagle fastIBD with the fastIBD score of 1e-10 and 5 independent runs of Beagle IBD with ibd2nonibd=0.01 and nonibd2ibd=0.0001 (optional, default: FDRref.txt)

-t, --fdr: the cutoff of FDR (optional, default: 0.1)

Outputs for ExIBD_Filtered:

The final IBD output file with the suffix 'exIBD' is produced after the implementation of ExIBD_Filtered. It reports the filtered IBD segments shared by pairs of samples detected by exome sequencing data under the specified FDR. This file has seven columns. The first two columns list the two sample identifier for the shared IBD segment. The third column lists the chromosomoe. The fourth and fifth columns list the starting (inclusive) and ending (inclusive) marker physical position for the shared IBD segment. The sixth and seventh columns list the starting (inclusive) and ending (inclusive) marker genetic position for the shared IBD segment.

4.4 Example

Here is an example which shows how to detect IBD segments in the given example dataset step by step.

1) Copy all the executable files and reference file in the directory ./bin into the current working directory ./example

   $ cp ../bin/* ./

2) Run Beagle fastIBD ten times with different random seeds, and set the fastIBD score threshold as 1e-10. If desired, this process can be conducted by using multiple threads. After running Beagle fastIBD, ten fastIBD files will be produced, such as 0.example.chr19.fibd.gz, 1.example.chr19.fibd.gz, …, and 9.example.chr19.fibd.gz,.

   $ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=0 seed=537
   $ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=1 seed=463132
   $ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=2 seed=1236
   $ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=3 seed=807
   $ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=4 seed=235312
   $ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=5 seed=673421

$ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=6 seed=76765
$ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=7 seed=23645
$ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=8 seed=564
$ java -Xmx3000m -jar beagle.jar unphased=example.chr19.gt missing=0 fastibd=true fastibdthreshold=1e-10 out=9 seed=7641

3) Combine the IBD calls from the 10 runs of Beagle fastIBD. Two files are generated here, i.e., example.chr19.gt.fibd and example.chr19.gt.candidate.

   $ python ./ExIBD_Candidate.py --file example.chr19.gt --round 10

   or

   $ ./ExIBD_Candidate --file example.chr19.gt --round 10

4) Refine the endpoints of candidate IBD segments by 5 independent runs of Beagle IBD. One file (i.e., example.chr19.gt.ribd) is generated at this step.

   $ ./ExIBD_Refined --file example.chr19.gt --numMarker 500 --numFIBD 20 --round 5 --beagle "java -Xmx1024m -jar beagle.jar unphased=example.chr19.gt missing=0" --threads 3

5) Filter based on the pre-calculated locus specific FDR. The final IBD file (i.e., example.chr19.gt.exIBD) is generated at this step.

   $ python ./ExIBD_Filtered.py --file example.chr19.gt --chr 19

   or

   $ ./ExIBD_Filtered --file example.chr19.gt --chr 19

5. Getting Started: running the pipeline

Before running ExIBD as a pipeline, please copy all the executable files (i.e., beagle.jar, ExIBD, ExIBD_Candidate, ExIBD_Refined, ExIBD_Filtered), and FDRref.txt into the same directory where the input files exist.

5.1 The pipeline

Inputs for ExIBD:

The genotypes file, the markers file, and the samples file are required.

Command line for ExIBD:

$ ./ExIBD --file <filename> --numMarker <number of markers> --chr <No. chromosome> --beagle <basic commands for Beagle> [OPTIONS]

Arguments for ExIBD:

-h, --help: print this help and exit

-f, --file: the full name of the genotypes file, required

-l, --numMarker: the number of markers, required

-c, --chr: the chromosome name, required

-b, --beagle: the basic command for Beagle, where the path of java, the memory requirement for both Beagle fastIBD and IBD, and some arguments for Beagle (such as unphased, phased, trios, pairs, like, missing, maxlr, niterations, nsamples, gprobs, lowmem, excludecolumns, nimputations, redundant) can be set here. **Note that double quotation marks are required for this command**, for example, --beagle "java –Xmx3000m –jar beagle.jar unphased=example.chr19.gt missing=0"; required

-r, --roundFastIBD: the number of Beagle fastIBD runs (optional, default: 10)

-i, --roundIBD: the number of Beagle IBD runs (optional, default: 5)

-e, --extend: the number of markers extended from the breakpoints of candidate IBD segments during the refinement process (optional, default: 0)

-d, --fastibdthreshold: fastIBD score threshold that controls fastIBD output (optional, default: 1e-10)

-u, --ibd2nonibd: the transition rate from IBD to non-IBD per cM for each sample (optional, default: 0.01)

-v, --nonibd2ibd: the transition rate from non-IBD to IBD per cM for each sample (optional, default: 0.0001)

-w, --ibderror: the estimated genotype error rate when estimating IBD (optional, default: 0.005)

-s, --ibdscale: a tuning parameter that controls the complexity of the haplotype frequency model when performing IBD analysis (optional, default: 2.0)

-p, --cutoff: the cutoff of IBD probability for each position during the calling of IBD segments (optional, default: 0.5)

-a, --reference: the reference file for the pre-assigned locus specific FDRs. Note that the default reference file is based on 10 independent runs of Beagle fastIBD with the fastIBD score of 1e-10 and 5 independent runs of Beagle IBD with ibd2nonibd=0.01 and nonibd2ibd=0.0001 (optional, default: FDRref.txt)

-g, --fdr: the cutoff of FDR (optional, default: 0.1)

-t, --threads: the number of threads of execution (optional, default: 1)

Outputs for ExIBD:

The fastIBD output files ('.fibd', '.candidate'), the refined IBD output file ('.ribd') and the final IBD output file ('.exIBD') are produced after the implementation of ExIBD. The output files from Beagle fastIBD are also kept.

5.2 Example

Here is an example which shows how to detect IBD segments in the given example dataset as a pipeline.

1) Copy all the executable files and reference file in the directory ./bin into the current working directory ./example

   $ cp ../bin/* ./

2) Run ExIBD

   $ ./ExIBD --file example.chr19.gt --numMarker 500 --chr 19 --roundFastIBD 10 --roundIBD 5 --beagle "java –Xmx3000m –jar beagle.jar unphased=example.chr19.gt missing=0" --threads 3