STITCH - Sequencing To Imputation Through Constructing Haplotypes

Current Version: 1.5.3
Release date: September 7th, 2018

Changes in latest version

1. Significant speedups and some RAM improvements
2. Minor changes to heuristics

For details of past changes please see CHANGELOG.

STITCH is an R program for reference panel free, read aware, low coverage sequencing genotype imputation. STITCH runs on a set of samples with sequencing reads in BAM format, as well as a list of positions to genotype, and outputs imputed genotypes in VCF format.

For the old website, please see http://www.well.ox.ac.uk/~rwdavies/stitch.html

Installation and quick start on real data example

Quick start on Linux and Mac

Install R if not already installed. Then

```
git clone --recursive https://github.com/rwdavies/STITCH.git
cd STITCH
./scripts/install-dependencies.sh
R CMD INSTALL ./releases/STITCH_1.5.3.tar.gz

# test on CFW mouse data
wget http://www.well.ox.ac.uk/~rwdavies/ancillary/STITCH_example_2016_05_10.tgz
# or curl -O http://www.well.ox.ac.uk/~rwdavies/ancillary/STITCH_example_2016_05_10
tar -xzvf STITCH_example_2016_05_10.tgz
./STITCH.R --chr=chr19 --bamlist=bamlist.txt --posfile=pos.txt --genfile=gen.txt --
# if this works the file stitch.chr19.vcf.gz will be created
```

If you see an error similar to `error while loading shared libraries: libmpc.so.2: cannot open shared object file: No such file or directory`, then either ask your system administrator to install gmp, mpfr and mpc for you, or try running the following before R CMD INSTALL

```
./scripts/install-package-dependencies.sh
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:`pwd`/install/lib/
```

If you see an error similar to `configure: error: liblzma not found, please install lzma`, then either ask your system administrator to install lzma or xz for you, or try running the following before R CMD INSTALL

```
./scripts/install-xz.sh
echo "CPPFLAGS += -I`pwd`/install/include" >> ~/.R/Makevars
echo "LDFLAGS += -I`pwd`/install/lib" >> ~/.R/Makevars
```

If you're on Mac you may see an error similar to `ld: library not found for -lquadmath`, which is related to STITCH C++ compilation using Rcpp. This can be fixed by updating gfortran using a method such as this. If you experience other compilation issues, please raise an issue. To experiment with configuration options during compilation, you can edit `STITCH/src/Makevars` then build a package and install using `./scripts/build-and-install.sh` or test using `./scripts/test-unit.sh`.

Interactive start

1. Install R if not already installed.
2. Install R dependencies parallel, Rcpp and RcppArmadillo from CRAN (using the "install.packages" option within R)
3. Install bgzip and make it available to your PATH. This can be done using a system installation, or doing a local installation and either modifying the PATH variable using code like `export PATH=/path/to/dir-with-bgzip-binary/:$PATH`, or through R, doing something like `Sys.setenv( PATH = paste0("/path/to/dir-with-bgzip-binary/:", Sys.getenv("PATH")))`. You'll know samtools is available if you run something like `system("which bgzip")` in R and get the path to bgzip
4. Install STITCH. First, download the latest STITCH tar.gz from the releases folder above. Second, install by opening R and using install.packages, giving install.packages the path to the downloaded STITCH tar.gz. This should install SeqLib automatically as well.
5. Download example dataset STITCH_example_2016_05_10.tgz.
6. Run STITCH. Open R, change your working directory using setwd() to the directory where the example tar.gz was unzipped, and then run `STITCH(tempdir = tempdir(), chr =`

`"chr19", bamlist = "bamlist.txt", posfile = "pos.txt", genfile = "gen.txt", outputdir = paste0(getwd(), "/"), K = 4, nGen = 100, nCores = 1)`. Once complete, a VCF should appear in the current working directory named stitch.chr19.vcf.gz

Help, command line interface and common options

For a full list of options, in R, query `?STITCH`, or from the command line, `STITCH --help`. For a brief writeup of commonly used variables, see Options.md. To pass vectors using the command line, do something like `STITCH.R --refillIterations='c(3,40)'` or `STITCH.R --reference_populations='c("CEU","GBR")'`.

Benchmarks

One can see some speed benchmarks in benchmarks/summarize_benchmarking.md

Examples

In the examples directory, there is a script which contains examples using real mouse and human data. One can either run this interactively in R, or run all examples using `./examples/example.R`.

License

STITCH and the code in this repo is available under a GPL3 license. For more information please see the LICENSE.

Testing

Tests in STITCH are split into unit or acceptance run using `./scripts/test-unit.sh` and `./scripts/test-acceptance.sh`. To run all tests use `./scripts/all-tests.sh`, which also builds and installs a release version of STITCH. To make compilation go faster do something like `export MAKE="make -j 8"`.

Citation

Davies, R. W., Flint J, Myers S., Mott R. Rapid genotype imputation from sequence without reference panels. *Nat. Genet.* 48, 965-969 (2016)

Contact and bug reports

The best way to get help is to either submit a bug report on GitHub or to consult the forum and mailing list

https://groups.google.com/forum/#!forum/stitch-imputation

For more detailed questions or other concerns please contact Robert Davies robertwilliamdavies@gmail.com

Output format

STITCH supports writing to both bgzipped vcfs and bgen, see output_format variable

What method to run

STITCH can run using one of three "methods" reflecting different underlying statistical and biological models: "diploid", which is the best general method and has the best statistical properties, but has run time proportional to the square of K and so may be slow for large, diverse populations; "pseudoHaploid", which uses statistical approximations that make it less accurate than the diploid method but has run time proportional to K, and so may be suitable for large, diverse populations; and "diploid-inbred", which assumes all samples are completely inbred and as such uses an underlying haplotype based imputation model with run time proportional to K. Note that each of these assumes subjects are diploid, and as such, all methods output diploid genotypes and probabilities.

Notes on the relationship between run time, RAM and performance

STITCH can be run on hundreds of thousands of samples, SNPs, or both. Default parameters are set to give good performance for situations somewhere in the middle. Depending on your application, you may want to tweak default parameters to change how STITCH is run and the relationship between run time, RAM and performance. Here is a brief summary of relevant parameters. See section below for note about K.

- outputSNPBlockSize: STITCH writes out results approximately this many SNPs at a time. Setting this to a larger value will speed up STITCH but use more RAM.
- keepSampleReadsInRAM, inputBundleBlockSize: STITCH converts reads from BAM files into an internal format. These variables control whether all of those are kept in RAM

(keepSampleReadsInRAM = TRUE) or not (keepSampleReadsInRAM=FALSE, default) at once. Setting to TRUE decreases runtime but increases RAM usage. inputBundleBlockSize controls whether sampleReads are bundled together to use fewer temporary files on disk. These options have no impact on performance.

- gridWindowSize: The default gridWindowSize=NA makes imputation run per-SNP, as is standard. Setting this to an integer greater than 0 (e.g. 10000 (base pairs)) bins the genome into physical windows of this size, and runs imputation between those grids. This can considerably speed up imputation of dense regions but will reduce imputation performance.

Note on the selection of K and nGen

A fuller description is given the supplement of the paper given in the citation, and this is worth a read for anyone planning to use the method in their work.

K is the number of ancestral haplotypes in the model. Larger K allows for more accurate imputation for large samples and coverages, but takes longer and accuracy may suffer with lower coverage. It is usually wise to try a few values of K and assess performance using either external validation, or the distribution of quality scores (e.g. mean / median INFO score). It is likely wise to choose K that both gives you the best performance (accuracy, correlation or quality score distribution) within computational constraints, while also ensuring K is not too large given your sequencing coverage (e.g. try to ensure that each ancestral haplotype gets at least a certain average X of coverage, say 10X, given your number of samples and average depth).

nGen controls recombination rate between the sequenced samples and the ancestral haplotypes. It is probably fine to set it to 4 * Ne / K given some estimate of effective population size Ne. If you think your population can reasonably approximated as having been founded some number of generations ago / reduced to 2*K that many generations ago, use that generation time estimate. STITCH should be fairly robust to misspecifications of this parameter.