Package 'STITCH'

July 18, 2024

Type Package

Title STITCH - Sequencing To Imputation Through Constructing Haplotypes

Version 1.7.0

Date 2024-07-18

Author Robert William Davies

Maintainer Robert William Davies <robertwilliamdavies@gmail.com>

Description STITCH performs imputation of individuals sequenced to low coverage in a read aware fashion without a reference panel.

Installation To install, first install dependencies, then run the install.packages command, pointing to the downloaded tarball (STITCH.tar.gz)

Getting started A minimum run requires the following options to be set: the chromosome being run (chr); a path to a file with a set of bi-allelic SNP sites (posfile); a choice of K, the number of internally modelled haplotypes (K); a path to an output directory (outputdir); a path to a temporary directory, ideally on fast disks or a RAM disk (tempdir); a list of bam files (bamlist); and the number of generations since founding (nGen), which can be approximated from a choice of K for wild populations from 4 * Ne / K. Additional useful options relate to what region to impute (regionStart, regionEnd, buffer), whether to use validation data to benchmark imputation (genfile), the number of cores to use (nCores), whether imputation is run on a server or cluster (environment), the number of EM iterations (niterations), whether to run in diploid or pseudoHaploid mode (method), and if run in pseudoHaploid mode, what iteration to switch from pseudoHaploid to diploid (switchModelIteration).

Depends parallel

Imports Rcpp, data.table

Suggests testthat, rrbgen (>= 0.0.4)

Remotes github::rwdavies/rrbgen/rrbgen

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.1

License GPL | file LICENSE

SystemRequirements C++11

NeedsCompilation yes

Contents

`extract_hd_to_cube` *Extract ancestral haplotype dosage to RData cube*

Description

Extract ancestral haplotype dosage to RData cube

Usage

```
extract_hd_to_cube(
  vcf_file,
  ref,
  bcftools = "bcftools",
  gatk_jar =
   "/data/smew1/rdavies/stitch_richard_paper/bin/gatk/GenomeAnalysisTK-3.8-1-0-gf15c1c3ef/GenomeAnal
  samples = NULL,
  pos = NULL,
  ram = "-Xmx4g",
  field = "HD",
  verbose = TRUE
)
```

Arguments

`vcf_file`	path to VCF
`ref`	path to reference fasta (required by GATK)
`bcftools`	path to vcftools (or just "bcftools" if in path)
`gatk_jar`	path to GATK jar
`samples`	vector of sample names (or NULL for all)

pos	pos matrix, a matrix with at least two columns where the first two columns are chrom and 1-based physical position, respectively. specifies which SNPs to extract (or NULL for all)
field	What to get from the VCF. Default HD for haplotype dosages

Value

A cube with dimensions of SNPs x samples x ancestral haplotypes

Author(s)

Robert Davies

---

make_STITCH_cli *Make STITCH command line interface*

Description

Make STITCH command line interface

Usage

```
make_STITCH_cli(
  function_file,
  cli_output_file,
  integer_vectors = c("shuffleHaplotypeIterations", "splitReadIterations",
    "refillIterations", "reference_shuffleHaplotypeIterations"),
  character_vectors = c("reference_populations"),
  other_logical_params = NULL,
  other_integer_params = NULL,
  other_double_params = NULL,
  other_character_params = NULL,
  function_name = "STITCH",
  library_name = "STITCH"
)
```

Arguments

function_file	to main STITCH function file
stitch_cli_file	where output goes

STITCH *Sequencing To Imputation Through Constructing Haplotypes*

Description

Sequencing To Imputation Through Constructing Haplotypes

Usage

```
STITCH(
  chr,
  nGen,
  posfile,
  K,
  S = 1,
  outputdir,
  nStarts,
  tempdir = NA,
  bamlist = "",
  cramlist = "",
  sampleNames_file = "",
  reference = "",
  genfile = "",
  method = "diploid",
  output_format = "bgvcf",
  B_bit_prob = 16,
  outputInputInVCFFormat = FALSE,
  downsampleToCov = 50,
  downsampleFraction = 1,
  readAware = TRUE,
  chrStart = NA,
  chrEnd = NA,
  regionStart = NA,
  regionEnd = NA,
  buffer = NA,
  maxDifferenceBetweenReads = 1000,
  maxEmissionMatrixDifference = 1e+10,
  alphaMatThreshold = 1e-04,
  emissionThreshold = 1e-04,
  iSizeUpperLimit = as.integer(600),
  bqFilter = as.integer(17),
  niterations = 40,
  shuffleHaplotypeIterations = c(4, 8, 12, 16),
  splitReadIterations = 25,
  nCores = 1,
  expRate = 0.5,
  maxRate = 100,
```

```
  minRate = 0.1,
  Jmax = 1000,
  regenerateInput = TRUE,
  originalRegionName = NA,
  keepInterimFiles = FALSE,
  keepTempDir = FALSE,
  outputHaplotypeProbabilities = FALSE,
  switchModelIteration = NA,
  generateInputOnly = FALSE,
  restartIterations = NA,
  refillIterations = c(6, 10, 14, 18),
  downsampleSamples = 1,
  downsampleSamplesKeepList = NA,
  subsetSNPsfile = NA,
  useSoftClippedBases = FALSE,
  outputBlockSize = 1000,
  outputSNPBlockSize = 10000,
  inputBundleBlockSize = NA,
  genetic_map_file = "",
  reference_haplotype_file = "",
  reference_legend_file = "",
  reference_sample_file = "",
  reference_populations = NA,
  reference_phred = 20,
  reference_iterations = 40,
  reference_shuffleHaplotypeIterations = c(4, 8, 12, 16),
  output_filename = NULL,
  initial_min_hapProb = 0.2,
  initial_max_hapProb = 0.8,
  regenerateInputWithDefaultValues = FALSE,
  plotHapSumDuringIterations = FALSE,
  plot_shuffle_haplotype_attempts = FALSE,
  plotAfterImputation = TRUE,
  save_sampleReadsInfo = FALSE,
  gridWindowSize = NA,
  shuffle_bin_nSNPs = NULL,
  shuffle_bin_radius = 5000,
  keepSampleReadsInRAM = FALSE,
  useTempdirWhileWriting = FALSE,
  output_haplotype_dosages = FALSE,
  use_bx_tag = TRUE,
  bxTagUpperLimit = 50000
)
```

Arguments

chr	What chromosome to run. Should match BAM headers
nGen	Number of generations since founding or mixing. Note that the algorithm is

relatively robust to this. Use nGen = 4 * Ne / K if unsure

`posfile` Where to find file with positions to run. File is tab seperated with no header, one row per SNP, with col 1 = chromosome, col 2 = physical position (sorted from smallest to largest), col 3 = reference base, col 4 = alternate base. Bases are capitalized. Example first row: 1<tab>1000<tab>A<tab>G<tab>

`K` How many founder / mosaic haplotypes to use

`S` How many sets of founder / mosaic haplotypes to use

`outputdir` What output directory to use

`tempdir` What directory to use as temporary directory. If set to NA, use default R tempdir. If possible, use ramdisk, like /dev/shm/

`bamlist` Path to file with bam file locations. File is one row per entry, path to bam files. Bam index files should exist in same directory as for each bam, suffixed either .bam.bai or .bai

`cramlist` Path to file with cram file locations. File is one row per entry, path to cram files. cram files are converted to bam files on the fly for parsing into STITCH

`sampleNames_file`
Optional, if not specified, sampleNames are taken from the SM tag in the header of the BAM / CRAM file. This argument is the path to file with sampleNames for samples. It is used directly to name samples in the order they appear in the bamlist / cramlist

`reference` Path to reference fasta used for making cram files. Only required if cramlist is defined

`genfile` Path to gen file with high coverage results. Empty for no genfile. File has a header row with a name for each sample, matching what is found in the bam file. Each subject is then a tab seperated column, with 0 = hom ref, 1 = het, 2 = hom alt and NA indicating missing genotype, with rows corresponding to rows of the posfile. Note therefore this file has one more row than posfile which has no header

`method` How to run imputation - either diploid, pseudoHaploid, or diploid-inbred. Please see main README for more information. All methods assume diploid samples. diploid is the most accurate but slowest, while pseudoHaploid may be advantageous for large sample sizes and K. diploid-inbred assumes all samples are inbred and invokes an internal haploid mathematical model but outputs diploid genotypes and probabilities

`output_format` one of bgvcf (i.e. bgziped VCF) or bgen (Layout = 2, CompressedSNPBlocks = 1)

`B_bit_prob` when using bgen, how many bits to use to store each double. Optiosn are 8, 16, 24 or 32

`outputInputInVCFFormat`
Whether to output the input in vcf format

`downsampleToCov`
What coverage to downsample individual sites to. This ensures no floating point errors at sites with really high coverage

`downsampleFraction`
: Downsample BAMs by choosing a fraction of reads to retain. Must be value 0<downsampleFraction<1

`readAware`
: Whether to run the algorithm is read aware mode. If false, then reads are split into new reads, one per SNP per read

`chrStart`
: When loading from BAM, some start position, before SNPs occur. Default NA will infer this from either regionStart, regionEnd and buffer, or posfile

`chrEnd`
: When loading from BAM, some end position, after SNPs occur. Default NA will infer this from either regionStart, regionEnd and buffer, or posfile

`regionStart`
: When running imputation, where to start from. The 1-based position x is kept if regionStart <= x <= regionEnd

`regionEnd`
: When running imputation, where to stop.

`buffer`
: Buffer of region to perform imputation over. So imputation is run form regionStart-buffer to regionEnd+buffer, and reported for regionStart to regionEnd, including the bases of regionStart and regionEnd

`maxDifferenceBetweenReads`
: How much of a difference to allow the reads to make in the forward backward probability calculation. For example, if P(read | state 1)=1 and P(read | state 2)=1e-6, re-scale so that their ratio is this value. This helps prevent any individual read as having too much of an influence on state changes, helping prevent against influence by false positive SNPs

`maxEmissionMatrixDifference`
: Similar to maxDifferenceBetweenReads, specifies ratio of how much larger the most probable state can be than the least probable state, but across all reads rather than for a single read. This helps to limit overflow in C++ calculations

`alphaMatThreshold`
: Minimum (maximum is 1 minus this) state switching into probabilities

`emissionThreshold`
: Emission probability bounds. emissionThreshold < P(alt read | state k) < (1-emissionThreshold)

`iSizeUpperLimit`
: Do not use reads with an insert size of more than this value

`bqFilter`
: Minimum BQ for a SNP in a read. Also, the algorithm uses bq<=mq, so if mapping quality is less than this, the read isnt used

`niterations`
: Number of EM iterations.

`shuffleHaplotypeIterations`
: Iterations on which to perform heuristic attempt to shuffle founder haplotypes for better fit. To disable set to NA.

`splitReadIterations`
: Iterations to try and split reads which may span recombination breakpoints for a better fit

`nCores`
: How many cores to use

`expRate`
: Expected recombination rate in cM/Mb

`maxRate`
: Maximum recomb rate cM/Mb

`minRate` Minimum recomb rate cM/Mb

`Jmax` Maximum number of SNPs on a read

`regenerateInput`
Whether to regenerate input files. If this is FALSE, please using the same regionStart, regionEnd, buffer and posfile as you used to generate the input. Setting any of those to different values can cause the previous input data to be improperly interpreted. Please also see originalRegionName and regenerateInputWithDefaultValues

`originalRegionName`
If regenerateInput is FALSE (i.e. using existing data), this is the name of the original region name (chr.regionStart.regionEnd). This is necessary to load past variables

`keepInterimFiles`
Whether to keep interim parameter estimates

`keepTempDir` Whether to keep files in temporary directory

`switchModelIteration`
Whether to switch from pseudoHaploid to diploid and at what iteration (NA for no switching)

`generateInputOnly`
Whether to just generate input data then quit

`restartIterations`
In pseudoHaploid method, which iterations to look for collapsed haplotype prnobabilities to resolve

`refillIterations`
When to try and refill some of the less frequently used haplotypes

`downsampleSamples`
What fraction of samples to retain. Useful for checking effect of N on imputation. Not meant for general use

`downsampleSamplesKeepList`
When downsampling samples, specify a numeric list of samples to keep

`subsetSNPsfile` If input data has already been made for a region, then subset down to a new set of SNPs, as given by this file. Not meant for general use

`useSoftClippedBases`
Whether to use (TRUE) or not use (FALSE) bases in soft clipped portions of reads

`outputBlockSize`
How many samples to write out to disk at the same time when making temporary VCFs that are later pasted together at the end to make the final VCF. Smaller means lower RAM footprint, larger means faster write.

`outputSNPBlockSize`
How many SNPs to write to disk at one time to reduce RAM usage when making VCFs

`inputBundleBlockSize`
If NA, disable bundling of input files. If not NA, bundle together input files in sets of <= inputBundleBlockSize together

genetic_map_file
: Path to file with genetic map information, a file with 3 white-space delimited entries giving position (1-based), genetic rate map in cM/Mbp, and genetic map in cM

reference_haplotype_file
: Path to reference haplotype file in IMPUTE format (file with no header and no rownames, one row per SNP, one column per reference haplotype, space separated, values must be 0 or 1)

reference_legend_file
: Path to reference haplotype legend file in IMPUTE format (file with one row per SNP, and a header including position for the physical position in 1 based coordinates, a0 for the reference allele, and a1 for the alternate allele)

reference_sample_file
: Path to reference sample file (file with header, one must be POP, corresponding to populations that can be specified using reference_populations)

reference_populations
: Vector with character populations to include from reference_sample_file e.g. CHB, CHS

reference_phred
: Phred scaled likelihood or an error of reference haplotype. Higher means more confidence in reference haplotype genotypes, lower means less confidence

reference_iterations
: When using reference haplotypes, how many iterations to use to train the starting data

reference_shuffleHaplotypeIterations
: When using reference haplotypes, how much shuffling to do to lead to better global fit

output_filename
: Override the default bgzip-VCF / bgen output name with this given file name. Please note that this does not change the names of inputs or outputs (e.g. RData, plots), so if outputdir is unchanged and if multiple STITCH runs are processing on the same region then they may over-write each others inputs and outputs

initial_min_hapProb
: Initial lower bound for probability read comes from haplotype. Double bounded between 0 and 1

initial_max_hapProb
: Initial upper bound for probability read comes from haplotype. Double bounded between 0 and 1

regenerateInputWithDefaultValues
: If regenerateInput is FALSE and the original input data was made using regionStart, regionEnd and buffer as default values, set this equal to TRUE

plotHapSumDuringIterations
: Boolean TRUE/FALSE about whether to make a plot that shows the relative number of individuals using each ancestral haplotype in each iteration

plot_shuffle_haplotype_attempts
: Boolean TRUE/FALSE about whether to make a plot that tries to show the selection of ancestral haplotypes to check for shuffling / flipping

`plotAfterImputation`
: Boolean TRUE/FALSE about whether to make plots after imputation has run (can be set to FALSE if this throws errors on systems without x11)

`save_sampleReadsInfo`
: Experimental. Boolean TRUE/FALSE about whether to save additional information about the reads that were extracted

`gridWindowSize`
: Whether to work on a grid where reads are binned into windows of this size (1 based, i.e. first bin is bases 1-gridWindowSize). This is particularly appropriate for very low coverage data (e.g. less than 0.2X) and can substantially speed up analyses

`shuffle_bin_nSNPs`
: Parameter that controls how to detect ancestral haplotypes that are shuffled during EM for possible re-setting. If set (not NULL), then break per-SNP (or per-grid) every this many SNPs / grids, and compare each to detect whether haplotypes either 1) are more likely to stay where they are or 2) switch from one haplotype to another. Note that only one of shuffle_bin_nSNPs or shuffle_bin_radius should be non-NULL

`shuffle_bin_radius`
: Parameter that controls how to detect ancestral haplotypes that are shuffled during EM for possible re-setting. If set (not NULL), then recombination rate is calculated around pairs of SNPs in window of twice this value, and those that exceed what should be the maximum (defined by nGen and maxRate) are checked for whether they are shuffled

`keepSampleReadsInRAM`
: Whether to (generally) keep sampleReads in RAM or store them in the temporary directory. STITCH is substantially faster if this is FALSE at the expense of RAM

`useTempdirWhileWriting`
: Whether to use temporary directory while writing output file (TRUE), or to keep result in RAM (FALSE). Using temporary directory is slower but uses less RAM

`output_haplotype_dosages`
: Whether to output ancestral haplotype dosages, i.e. the expected number of ancestral haplotypes carried by that sample at that locus

`use_bx_tag`
: Whether to try and use BX tag in same to indicate that reads come from the same underlying molecule

`bxTagUpperLimit`
: When using BX tag, at what distance between reads to consider reads with the same BX tag to come from different molecules

Value

Results in properly formatted version

Author(s)

Robert Davies

Index