Instruction Manual for “SOURCEFINDv2: a program for inferring proportions of haplotype sharing among populations”

Garrett Hellenthal

February 10, 2019

Contents

1 Introduction

SOURCEFINDv2 is an R program, originally described in [1], that infers the proportion of DNA for which each “target” individual (or group of individuals) shares recent ancestry with a set of “surrogate” individuals. For example, these surrogates may represent possible ancestral source groups from which the target group may descend. The input for SOURCEFINDv2 is the total genomewide amount of DNA for which each “target” and “surrogate” group or individual shares a most recent common ancestor with a set of "donor" groups or individuals, based on haplotype sharing as inferred by the program CHROMOPAINTER [2]. Here donors can be e.g. the same surrogate and target groups, or a completely different set of groups. SOURCEFINDv2 then infers how best to represent the target group's inferred haplotype sharing patterns as a mixture of that from the surrogate groups, providing the mixture coefficients as output. SOURCEFINDv2 is similar to the mixture model approach of GLOBETROTTER described in [3, 4], though using a different model that includes a Markov-Chain-Monte-Carlo (MCMC) procedure and that showed improved performance in simulations when e.g. used to identify admixture proportions in Latin Americans [1]. (However, we caution that inferred proportions need not be at all related to admixture; instead SOURCEFINDv2 measures relative amounts of shared recent ancestry between the target and the set of surrogates included in the mixture.)

2 Running CHROMOPAINTER to make input files for SOURCEFINDv2

SOURCEFINDv2 uses the XX.chunklengths.out (or XX.chunkcounts.out) files generated by the program CHROMOPAINTER [2], which identifies haplotype sharing patterns among different groups by studying Single-Nucleotide-Polymorphism (SNP) data. Specifically, CHROMOPAINTER "paints" the DNA of a "recipient" population conditional on a set of "donor" populations. I.e. at each genetic region across the haplotypes of a set of recipient individuals, CHROMOPAINTER identifies the single best matching haplotype among the set of donor individuals, which is indicative that the recipient and donor share recent common ancestry at the particular genetic locus.

In particular, the target and surrogate populations should copy from the same set of donors under the CHROMOPAINTER model. As noted above, this set of donors could contain all individuals from the target and recipient populations, as in the "full" analyses described in [3]. Or the set of donors might contain only the surrogate populations, as is the case in the "regional analyses" of [3] and the analyses of [4] and [1]. However, the set of donors could also be entirely separate from the surrogate and donor groups, or only contain a subset of these groups and/or a subset of individuals from these groups.

3 Getting Started

After extracting the .tar file, the basic command line is as follows:

R < sourcefindv2.R [*parameter_infile*] --no-save

or to direct screen output to another file "*screen_output*":

```
R < sourcefindv2.R [parameter_infile] --no-save > [screen_output]
```

There is one required user-input file, the *parameter_infile*.

Type "R < sourcefindv2.R help --no-save" to get a brief description of this command line and the parameter input file options.

4 Input Format

4.1 *parameter_infile*

There is a single input file (*parameter_infile*) that must be specified from the command line, which we have provided an example for with "example/BrahuiYorubaSimulation.SourcefindParamfile.txt". This example file reflects the simulated example in Figure 1 of [3].

The file contains all the parameter information for the SOURCEFINDv2 run, as well as the names of the files containing (a) all the copying vectors for the surrogate and target populations and (b) the labels and group assignments for all surrogate, target, and donor individuals. The file contains 13 rows, formatted in the following manner (and order) shown in bold type, with brackets containing allowed values and a description provided in normal font provided to the right of each parameter:

- **self.copy.ind:** [**0,1**] – indicate whether ("**1**") or not ("**0**") to allow "self-copying" in each target population. If allowed, this will infer the excess amount by which members of the target group match DNA patterns to other members of their own group label, beyond what can be explained by modelling the target as a mixture of the surrogates. Roughly speaking, this can be thought of as measuring the excess "drift" in the target group that cannot be explained by using the surrogate groups. (Note: only works if the target is included among the donors.)

- **num.surrogates:** [**2,...**] – number of surrogates that can be used to form the target group in each MCMC iteration

- **exp.num.surrogates:** [**1,...,num.surrogates**] – expected number of surrogates (mean of a Poisson prior) used to form the target group in each MCMC iteration

- **input.file.ids:** [**input.filename1**] – pathway and name for file containing id labels for all samples (see Section 4.2 below)

- **input.file.copyvectors:** [**input.filename2**] – pathway and name for file containing copy vectors for all surrogate and target populations (see Section 4.3 below)

- **save.file.main:** [**output.filename**] – pathway and name for the output file (see Section 5)

- **copyvector.popnames: [pop_1 pop_2 ... pop_k]** – names of all k populations used as donors; i.e. that both surrogate and target populations copied from when running CHROMOPAINTER

- **surrogate.popnames: [pop_1 pop_2 ... pop_j]** – names of all j surrogate populations, i.e. used to describe the ancestry in each of the target groups

- **target.popnames: [pop_rec1 pop_rec2 ...]** – name(s) of target population(s)

- **num.slots: [1,2,...]** – at each MCMC iteration, assign each of **num.slots** equally-sized proportions to one of the **num.surrogates** surrogates selected at that MCMC iteration

- **num.iterations: [1,2,...]** – total number of MCMC iterations

- **num.burnin: [1,2,...]** – discard the first **num.burnin** MCMC iterations as "burn-in"

- **num.thin: [1,2,...]** – after "burn-in", sample an MCMC iteration every **num.thin** iterations

In short, in each MCMC iteration, a maximum of **num.surrogates** surrogates (with an expected value of **exp.num.surrogates** surrogates) are allowed to contribute to the mixture model describing the target population(s), with each of **num.slots** equally-sized proportions of DNA from the target group divided among the **num.surrogates** chosen. For example, if **num.slots**=100, then each bin of proportion is worth 1% of the total proportion of the target, and any of the **num.surrogates** surrogates selected at the given MCMC iteration can claim each of these 100 bins. In contrast, if **num.slots**=200, each bin of proportion is worth 0.5% of the total proportion of the target, etc.

4.2 input.file.ids

This file should provide SOURCEFINDv2 the group labels for each individual specified in the rows and/or columns of **input.file.copyvectors**. (Note that if the rows/columns of **input.file.copyvectors** only contain labels provided in **copyvector.popnames**, **surrogate.popnames** and/or **target.popnames**, then this **input.file.ids** file is ignored, though it still must be specified.)

This file is in the same format as the ChromoPainterv2 '-t' switch, and should specify the individual identifier and corresponding population label for each donor, target, and surrogate individual in the analysis. The format of the file is one individual per row. There are three columns per row, with the first column giving the individual identifier, the second column giving the individual's population label and the third column an indicator for whether the individual is not included in the analysis (use "0" – i.e. "zero" – to specify NOT to include the given individual). Any individuals with a “0” in the third column will NOT be considered in any part of the SOURCEFINDv2 analysis.

For example, consider a file with the following 10 individuals:

```
IND1 Pop1 0
IND3 Pop1 1
IND2 Pop1 1
IND4 Pop2 1
IND5 Pop2 0
IND6 Pop2 1
IND7 Pop1 1
Pop4Ind1 Pop4 1
IND8 Pop3 1
IND9 Pop3 1
```

In the SOURCEFINDv2 analysis, IND1 and IND5 will be ignored. This leaves IND3, IND2, IND7 as representing “Pop1”, IND4, IND6 representing “Pop2”, IND8, IND9 representing “Pop3” and Pop4Ind1 representing “Pop4”. Therefore, SOURCEFINDv2 will search for these individual labels (i.e. column 1) when combining copy vector columns and rows from **input.file.copyvectors** for each of these four population labels “Pop1-Pop4”, though when performing this concatonation for each of “Pop1-Pop4” it will also include any columns and rows from **input.file.copyvectors** labeled as “Pop1-Pop4”, respectively. (Note that no value other than “0” in the third column specifies any action.)

An example of **input.file.ids** is provided in **“example/BrahuiYorubaSimulation.idfile.txt”**. Here column 3 has a “1” in each row, specifying that no individuals are removed from analysis.

4.3 input.file.copyvectors

This file should contain the XXX.chunklengths.out files from the corresponding preliminary CHROMOPAINTER analyses, in the same format and combined across all chromosomes and individuals. I.e. each row is an individual (or population), and the columns give the total amount of genome-wide DNA that the given individual (or population) is inferred to copy from every “donor” individual (or population) in the corresponding preliminary CHROMOPAINTER analyses.

The first row of **input.file.copyvectors** lists the column labels reflecting the donor individuals and/or populations. The first column in this first row is “Recipient”, and the remaining columns of this first row must contain for each label specified in **copyvector.popnames** of “*parameter_infile*” either (i) the label itself and/or (ii) the individual identifier of at least one individual assigned to that label, in any order (i.e. the order does NOT need to match that of

input.file.ids). (The individual identifiers and their corresponding labels are specified in columns 1 and 2, respectively, of **input.file.ids** – see Section 4.2.) The remaining rows of **input.file.copyvectors** list the "recipient" individual (or population) label in the first column, with the remaining columns containing the total amount (or proportion) of genome-wide DNA that the given recipient individual (or population) copies from each donor label provided in the first row. Analogous to the first row, the first column of **input.file.copyvectors**, which list the recipient labels, must contain for each label specified in **surrogate.popnames** and **target.popnames** of "*parameter_infile*" either the label itself and/or the individual identifier of at least one individual assigned to that label, again in any order.

An example of **input.file.copyvectors** is provided in **"example/BrahuiYorubaSimulation.copyvectors.txt"**. Here for column 1 we have used individual identifiers for all the recipients individuals (rows), while for row 1 we have simply used the population labels to specify each of the donors (columns). Therefore, for **target.popnames** and each label specified in **surrogate.popnames** of "*parameter_infile*", GLOBETROTTER will first find all individual identifiers (column 1 of **input.file.ids**) for the given label (column 2 of **input.file.ids**), pull out all rows of **input.file.copyvectors** matching these individual identifiers, and average each column across these rows to get the final matrix of copying vectors (i.e. without having to do any summing over columns in this particular example).

Note that you may include labels in row 1 and/or column 1 of **input.file.copyvectors** that are neither contained in **input.file.ids** nor correspond to any of **copyvector.popnames**, **surrogate.popnames** or **target.popname** of "*parameter_infile*". These rows/columns will be ignored in the SOURCEFINDv2 analysis. In addition, any rows/columns in **input.file.copyvectors** labeled with identifiers that have been specified to be excluded from analysis (i.e. by having a "0" in the third column of **input.file.ids**) will also be ignored.

5 Output

The output file, specified by **save.file** in "*parameter_infile*", contains a set of rows for each target group specified in **target.popnames**. There will be $M+1$ total rows for each of these target groups, with the first row corresponding to a header and the remaining rows corresponding to the M MCMC samples, where M=(**num.iterations - num.burnin**) / **num.thin**. Each of these rows has $S+2$ columns, where S is the number of surrogate groups specified in **surrogate.popnames**. (If **self.copy.ind**=1 and the given target group is not included in **surrogate.popnames** but is included in **copyvector.popnames**, then there will be an additional column corresponding to the target group.) The first column lists the name of the target group. The second column gives the posterior probability of that MCMC sample. The remaining columns give, for that MCMC sample, the inferred contributions to the mixture model from each of the surrogate groups specified in the header. For example, the mean across rows for each column could be used to give the final inferred proportion of ancestry for each surrogate group, as was used in [1].

If multiple target groups are specified in **target.popnames**, the $M+1$ rows of output for each will be stacked atop of each other.

6 Example of Usage

To run SOURCEFINDv2 on the provided example files, which uses as a target group the CHROMOPAINTER output for 20 individuals simulated as descendants of an admixture event occuring 30 generations ago, with 80% of the DNA contributed by the Brahui and 20% contributed by the Yoruba (this is the simulation described in Figure 1 of [3]), type the following:

```
R < sourcefindv2.R example/BrahuiYorubaSimulation.SourcefindParamfile.txt
--no-save > output.out
```

The output file "example/BrahuiYorubaSimulation.sourcefind.txt" will be generated. As **num.iteration: 200000**, **num.burnin: 50000**, and **num.thin: 5000** in "example/BrahuiYorubaSimulation.paramfile.txt", this output file will have 31 rows corresponding to the inferred proportions of ancestry for the single target group "BrahuiYorubaSimulation" analysed, i.e. with one header row and the rest corresponding to the $M = 30$ MCMC samples. Each row will have 95 columns, with the first two columns corresponding to the target group and posterior probabilities, respectively, and the remaining columns corresponding to the $S = 93$ groups specified in **surrogate.popnames**. For each MCMC sample (row), a maximum of 8 of these surrogates will have non-zero values, following the specification of **num.surrogates: 8**. For each of these contributing surrogates per MCMC sample, contributing values will be rounded to the nearest .01 (because **num.slots: 100**), with the total contributed proportion across all surrogates summing to 1.

7 Computational Complexity

The computational complexity of SOURCEFINDv2 is fairly minimal, most notably affected by the total number of MCMC iterations I and the specified number of surrogates per MCMC run (i.e. **num.surrogates**), though also by the number of surrogate groups S, number of donor groups and number of target groups. The memory requirements are the size of the matrix specified in the **input.file.copyvectors** file, plus $O(SM)$, where M is the total number of MCMC samples.

8 Computational Details (differences from SOURCEFINDv1.R)

SOURCEFINDv2.R uses the multinomial likelihood model described in [1], i.e. where the CHROMOPAINTER-inferred K-vector chunk-lengths of the target group are multinomially distributed with K parameters defined by a linear mixture of the (rescaled) K-vector chunk-lengths of the surrogate populations. The aim is to infer the coefficients of the linear mixture model using MCMC.

As described in [1], and tested in many (but not all) of the simulations and real data analyses described therein, SOURCEFINDv2.R puts a Poisson prior on the number of contributing surrogates (out of **num.surrogates** total). This Poisson prior has mean **exp.num.surrogates**.

For each MCMC step, SOURCEFINDv2.R uses the following proposal distribution, designed to efficiently and exhaustively sample the large space of possible mixing coefficients. As pointed out by Ida Moltke and Ryan Waples, the proposal density of SOURCEFINDv1.R was not symmetric and indeed is challenging to calculate. This version has some slight updates that makes the proposal symmetric, with these proposal updates being the only difference between SOURCEFINDv1.R and SOURCEFINDv2.R.

We first initially randomly pick a set S of Y(=**num.surrogates**) surrogates out of the J total surrogates available, and randomly assign each of the N (=**num.slots**) slots to one of the surrogates from S.

Then for each MCMC iteration:

1. Randomly select one surrogate out of S that is currently taking ≥ 1 slot, with probability equal to the proportion of slots that surrogate is currently taking. Call this surrogate S^*, which currently has $Z \leq N$ slots.
2. Then:
   (a) with probability 0.9:
      - randomly (uniformly) select $X = 1, ..., Z$ slots to change
      - if the total number of surrogates is $< Y$, randomly add surrogates to S (excluding S^* if $X = Z$) until there are Y surrogates to choose from
      - replace all of X with a single randomly selected surrogate (excluding S^*) from S

   (b) with probability 0.1, replace all of Z with a single randomly selected surrogate, excluding S^* and all surrogates surrently taking ≥ 1 slot

This proposal works as well as SOURCEFINDv1.R in all simulated applications I have applied it to, mainly the simulations that SOURCEFINDv1.R was applied to in [1]. However, finding an optimal proposal is a notoriously challenging problem in MCMC, so other updates could improve inference. As is good general practice with MCMC algorithms, trying multiple runs of the algorithm on your samples to assess consistency is important. Though this version mixes fairly well in all applications I have tested it on, this is on-going work. In general it is worth taking note of the uncertainty across MCMC samples, as well as assessing convergence, when reporting and interpreting results.

9 Citation

When making use of SOURCEFINDv2, please cite the following paper:

Chacon-Duque, J.C.; Adhikari, K.; Fuentes-Guajardo, M.; Mendoza-Revilla, J.; Acuna-Alonzo, V.; Lozano, R.B.; Quinto-Sanchez, M.; Gomez-Valdes, J.; Martinez, P.E.; Villamil-Ramirez, H.; Hunemeier, T.; Ramallo, V.; de Cerqueira, C.C.S.; Hurtado, M.; Villegas, V.; Granja, V.; Villena, M.; Vasquez, R.; Llop, E.; Sandoval, J.R.; Salazar-Granara, A.A.; Parolin, M.L.; Sandoval, K.; Penaloza-Espinosa, R.I.; Rangel-Villalobos, H.; Winkler, C.; Klitz, W.; Bravi, C.; Molina, J.; Corach, D.; Barrantes, R.; Gomes, V.; Resende, C.; Gusmao, L.; Amorim, A.; Xue, Y.; Dugoujon, J.M.; Moral, P.; Gonzalez-Jose, R.; Schuler-Faccini, L.; Salzano, F.M.; Bortolini, M.C.; Canizales-Quinteros, S.; Poletti, G.; Gallo, C.; Bedoya, G.; Rothhammer, F.; Balding, D.; Hellenthal, G.; Ruiz-Linares, A. (2018) "Latin Americans show wide-spread Converso ancestry and the imprint of local Native ancestry on physical appearance" *Nature Communications* 9:5388

When making use of ChromoPainter and/or fineSTRUCTURE, please cite the following paper:

Lawson, D.; Hellenthal, G.; Myers, S.; and Falush, D (2012) "Inference of population structure using dense haplotype data" *PLoS Genet* **8(1)**:e1002453

Questions? Bugs? Please contact Garrett Hellenthal at ghellenthal@gmail.com.

References

[1] Chacon-Duque, J.C. and Adhikari, K. and Fuentes-Guajardo, M. and Mendoza-Revilla, J. and Acuna-Alonzo, V. and Lozano, R.B. and Quinto-Sanchez, M. and Gomez-Valdes, J. and Martinez, P.E. and Villamil-Ramirez, H. and Hunemeier, T. and Ramallo, V. and de Cerqueira, C.C.S. and Hurtado, M. and Villegas, V. and Granja, V. and Villena, M. and Vasquez, R.

and Llop, E. and Sandoval, J.R. and Salazar-Granara, A.A. and Parolin, M.L. and Sandoval, K. and Penaloza-Espinosa, R.I. and Rangel-Villalobos, H. and Winkler, C. and Klitz, W. and Bravi, C. and Molina, J. and Corach, D. and Barrantes, R. and Gomes, V. and Resende, C. and Gusmao, L. and Amorim, A. and Xue, Y. and Dugoujon, J.M. and Moral, P. and Gonzalez-Jose, R. and Schuler-Faccini, L. and Salzano, F.M. and Bortolini, M.C. and Canizales-Quinteros, S. and Poletti, G. and Gallo, C. and Bedoya, G. and Rothhammer, F. and Balding, D. and Hellenthal, G. and Ruiz-Linares, A. Latin Americans show wide-spread Converso ancestry and the imprint of local Native ancestry on physical appearance. *Nature Communications*, 9(5388), 2018.

[2] D.J. Lawson, G. Hellenthal, S. Myers, and D. Falush. Inference of population structure using dense haplotype data. *PLoS Genet*, 8(1):e1002453, 2012.

[3] G. Hellenthal, G.B.J. Busby, G. Band, J.F. Wilson, C. Capelli, D. Falush, and S. Myers. A genetic atlas of human admixture history. *Science*, 343:747–751, 2014.

[4] S. Leslie, B. Winney, G. Hellenthal, D. Davison, A. Boumertit, T. Day, K. Hutnik, E.C. Royrvik, B. Cunliffe, Wellcome Trust Case Control Consortium 2, International Multiple Sclerosis Genetics Consortium, D.J. Lawson, D. Falush, C. Freeman, M. Pirinen, S. Myers, M. Robinson, P. Donnelly, and W. Bodmer. The fine scale genetic structure of the British population. *Nature*, 519:309–314, 2015.