# SeSAM: software for automatic construction of order-robust linkage maps

Adrien Vidal, Franck Gauthier, Willy Rodrigez, Nadège Guiglielmoni, Damien Leroux, Nicolas Chevrollier, Sylvain Jasson, Elise Tourrette, Olivier C. Martin, Matthieu Falque

# SeSAM: software for automatic construction of order-robust linkage maps

Adrien Vidal [1], Franck Gauthier [1], Willy Rodrigez [1], Nadège Guiglielmoni [1], Damien Leroux [2], Nicolas Chevrolier [1], Sylvain Jasson [2], Elise Tourrette [1], Olivier. C.Martin [1,3,4], and Matthieu Falque [1,*]

[1]Université Paris-Saclay, INRAE, CNRS, AgroParisTech, GQE - Le Moulon, 91190 Gif-sur-Yvette, France, [2]INRAE, Unité de Mathématiques et Informatique Appliquées - Toulouse, France, [3]Université Paris-Saclay, CNRS, INRAE, Université Evry, Institute of Plant Sciences Paris-Saclay (IPS2), 91190 Gif-sur-Yvette, France, [4]Université Paris Cité, CNRS, INRAE, Univ Evry, Institute of Plant Sciences Paris-Saclay (IPS2), 91190, Gif sur Yvette, France

[*] To whom correspondence should be addressed

Corresponding Author: matthieu.falque@inrae.fr

## Abstract

**Background**: Genotyping and sequencing technologies produce increasingly large numbers of genetic markers with potentially high rates of missing or erroneous data. Therefore, the construction of linkage maps is more and more complex. Moreover, the size of segregating populations remains constrained by cost issues and is less and less commensurate with the numbers of SNPs available. Thus, guaranteeing a statistically robust marker order requires that maps include only a carefully selected subset of SNPs.

**Results**: In this context, the SeSAM software allows automatic genetic map construction using seriation and placement approaches, to produce (1) a high-robustness *framework* map which includes as many markers as possible while keeping the order robustness beyond a given statistical threshold, and (2) a high-density *total* map including the framework plus almost all polymorphic markers. During this process, care is taken to limit the impact of genotyping errors and of missing data on mapping quality. SeSAM can be used with a wide range of biparental populations including from outcrossing species for which phases are inferred on-the-fly by maximum-likelihood during map elongation. The package also includes functions to simulate data sets, convert data formats, detect putative genotyping errors, visualize data and map quality (including graphical genotypes), and merge several maps into a consensus. SeSAM is also suitable for interactive map construction, by providing lower-level functions for 2-point and multipoint EM analyses. The software is implemented in a R package including functions in C++.

**Conclusions**: SeSAM is a fully automatic linkage mapping software designed to (1) produce a framework map as robust as desired by optimizing the selection of a subset of markers, and (2) produce a high-density map including almost all polymorphic markers. The software can be used with a wide range of biparental mapping populations including cases from

outcrossing. SeSAM is freely available under a GNU GPL v3 license and works on Linux, Windows, and macOS platforms. It is available as Additional file 1 and can be downloaded together with its user-manual and quick-start tutorial from ForgeMIA (SeSAM project) at https://forgemia.inra.fr/gqe-acep/sesam/-/releases

**Key-words:** genetic mapping, linkage, automated software, seriation, marker order robustness

## Background

Genetic linkage maps are representations of positions of polymorphic genetic elements along chromosomes, based on allele co-segregation patterns. Map distances are calculated from the frequency of meiotic crossovers between two linked loci; in the first historical maps, such frequencies were inferred from the segregation of phenotypes determined by two linked genes [1]. With the development of DNA technologies, the number of genetic markers increased, allowing genetic maps to become saturated, which means that any locus on the genome is significantly linked with at least one marker of the map [2, 3]. Linkage maps initially played an important role in unraveling the general organization of genomes [4], and in spite of genome sequencing becoming more and more accessible for structural genomics, they are still of great use e.g. for QTL detection via linkage or association studies, to help the orienting and placing of sequence contigs during genome assembly [5], or to detect errors *a posteriori* in assembled genomes [6].

In practice, genetic maps are built from observing the allelic segregation of polymorphic markers in mapping populations produced by different types of crossing schemes [7]. Biparental populations are the most frequently used, typically based on either two homozygous parents or two (partly) heterozygous parents as in the case of Cross-Pollinated

(CP) populations of many forest or fruit trees. The latter case involves more complex algorithms because current genotyping technologies do not directly provide long haplotype information, so the phase between multi-locus allelic configurations is unknown and must be inferred [8, 9]. Populations obtained from homozygous parents can be backcross (BC) or Doubled-Haploids (DH) which are very similar to BC with regards to map estimation, F2 to F*n*, Recombinant Inbred Lines (RIL) [10], or Intermated Recombinant Inbred lines (IRIL) populations. IRILs include some generations of random intermating between the F2 and the inbreeding generations, thereby increasing the number of crossovers captured and thus the resolution of the map for a given population size [11, 12].

The usual process for genetic map construction involves three successive steps [13] corresponding to (1) determination of linkage groups (when the map is saturated, linkage groups correspond to chromosomes), (2) ordering of markers in each linkage group, and (3) estimation of genetic distances between adjacent ordered markers. A lot of algorithmic effort has been made in particular for the ordering step, because as soon as the number of markers is not very small, it becomes unfeasible to evaluate an objective function for each possible order ($m!/2$ orders if $m$ is the number of markers). This problem, which is very similar to the Traveling Salesman Problem [14], is usually addressed in mapping softwares *via* different heuristics to escape this combinatorial explosion (see some examples in [13, 15–18] ; non-exhaustive list shown as Supplementary Table S1 in Additional file 2). The ordering algorithmic problem obviously becomes more difficult with recent genotyping technologies (including genotyping-by-sequencing) which can produce millions of SNPs. But with such technologies, an even more limiting issue is that whatever the algorithm, the information allowing ordering lies in the crossovers arising in the population, and thus scales up only with population size, which is generally much more expensive to increase than marker number. A

consequence is that if one wants to fix a minimum threshold for a robustness statistical criterion (for instance the minimum logarithm of odds (LOD) between the best order found and any other order), the number of markers in the map will be limited for a given population type and size: the higher the threshold, the lower the number of markers which can be included in the map. For usual levels of threshold (e.g. LOD=3) and large data sets, the maximum number of markers in the map will most often represent only a fraction of the SNPs available. The problem then translates into chosing the largest subset of markers which allows the order to be statistically robust at a given threshold. Here we propose the SeSAM (Seriation-based Suite for Automatic Mapping) package as a way to address the genetic mapping problem from this perspective.

Another consequence of the evolution of genotyping technologies is the number of missing data and/or genotyping errors, which can vary a lot depending on the approach used. For example, genotyping using low-coverage NGS sequencing [19–21] can produce many missing data which, depending on the protocol used for library preparation, can be distributed differently in the genome in different individuals of the mapping population. This is of particular concern for linkage analysis because detecting crossovers between two markers requires valid data in both markers. In multipoint estimations however, it is possible to impute part of the missing information for instance through Expectation-Maximization (EM) [22] algorithms, and it is possible to make use of data for genotype likelihoods [23], but beyond a certain level of missing data, map estimation always becomes challenging. The problem of genotyping errors is even more important when the number of markers becomes very large: each miscalled allele can produce a singleton interpreted as the result of two crossovers, thus for a given rate of genotyping errors, the more markers in the map, the more dramatically map length will be artificially inflated, and marker ordering altered [24]. A number of algorithms

identify singletons and putative erroneous data; replacing them by missing data limits their effect on the mapping outcome [25–28]. Conversely, it is also possible to identify markers that have a very low probability of displaying genotyping errors based on redundancy ("twins" approach [29–31]).

Numerous software tools have been developed for genetic mapping (see non-exhaustive list as Supplementary Table S1 in Additional file 2). Many of them feature sophisticated algorithms for marker ordering, some even include several different algorithms which can be compared to each other to assess the robustness of their outcome (see for instance [15] ). Most of the time, the main goal is to achieve optimal performances for finding the best order between all markers of a given linkage group (sometimes the $2^{nd}$, $3^{rd}$, etc... best maps are also provided). In some cases however, particularly (but not only) when population size is limited, an interesting alternative is the "bin-mapping" strategy [32–34]. In that approach, (1) a *framework* core skeletal map including only a subset of selected markers is produced to ensure an order robustness statistically supported at a desired threshold. The larger the mapping population, the more markers are included in this framework map. (2) Then all remaining polymorphic markers are *placed* within one of the bins delimited by the framework markers and their relative map position is calculated within the bin. Thus even though the order between close "placed" markers is not statistically supported, the order between each placed marker and the framework markers is. This strategy has several advantages: (1) the position of all markers can still be estimated precisely, while escaping the challenging computational problem of ordering too many markers, (2) the number of markers may become as high as desired, the computation time will remain close to linear with that number, (3) the uncertainty on the order of very close placed markers has no consequence on the estimated map length, and thus that uncertainty is no longer a problem for many applications.

In practice, such a bin-mapping strategy is usually carried out through an interactive process between an expert user and computer programs. Tools have been developed to automate the placing step [33], but to our knowledge, there is today no integrated software able to carry out a complete automated mapping process based on the bin-mapping strategy. So here we propose the SeSAM package, which automatically chains all steps necessary for genetic map construction based on this approach (Figure 1), the two main steps being: (1) producing a *framework* map by selecting an optimal subset of markers from the initial data set, so order robustness can be statistically supported, and (2) producing a *total* map by placing all remaining polymorphic markers one by one into that framework. In the first step, the determination of linkage groups is done during the elongation of an ordered low-density high-robustness map (*scaffold* map) through a seriation-based algorithm (Figure 1A) [35, 36], after which iterative densification of that scaffold leads to adding as many markers as possible while keeping the order robust at the desired threshold, which produces the *framework* map (Figure 1B). During this process, putative genotyping errors are detected and put aside to limit artifactual inflation of map length and ordering flaws. Finally, the remaining markers are *placed* on the framework (Figure 1C).

## Implementation

SeSAM is implemented as a R package, which allows to easily chain map construction to any other input data formatting (or map output exploiting) R script. The package includes C++ functions for its most computation-intensive parts (e.g. likelihood computations, EM algorithms). A detailed description of all algorithms and functions of SeSAM is provided in the user manual available as Additional file 3, and also from ForgeMIA at https://forgemia.inra.fr/gqe-acep/sesam/-/releases. The main functionality of SeSAM lies in

the function *autoMap()*, which is a completely automated pipeline going through different main steps carried out by the following functions: *loadData()*, which reads and checks segregation data, *generateSeeds()*, which draws the seed markers used to initiate the seriation process, *buildScaffold()*, which extends a highly robust sparse map by seriation from the seed markers, *assignment()*, which assigns all polymorphic markers to a linkage group, *buildFramework()*, which densifies the scaffolds with the maximum possible number of markers while keeping a given statistical level of order robustness to the framework map, and *placement()*, which adds all remaining markers to the framework map without ensuring a statistical value for order robustness. Missing data are imputed in all multipoint calculations via an EM algorithm, and putative genotyping errors are detected and taken into account *via* two different methods. Finally, the SeSAM package includes a toolbox of functions to perform various types of format conversions on data files, interactive step-by-step custom mapping processes, and assessment of map and data quality through different types of graphs. More detailed information about these different functions is available as Supplementary Text S1 in Additional File 2, and in the reference user manual available as Additional file 3 together with a quick-start guide.

## Results and Discussion

**Behavior with number of markers and population size** was assessed by simulating data sets for different population types, numbers of markers, and numbers of individuals. SeSAM was run on a desktop computer using 4 cores Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz (2 threads per core) under the Debian 11 Linux OS, using SeSAM default parameters. The scripts and data used to produce these benchmarking results are available in Additional file 4. Computation time for total maps construction was more or less linear with the number of

markers for F2 and CP populations (Figure 2). It was also close to linear with the number of individuals of the F2 population, but close to quadratic with the number of individuals for the CP population (Figure 3). Finally, Figure 4 shows that CP data sets necessitate substantially longer computation time than other population types, which is expected due to the extra phasing process required for such data.

To assess the quality of the maps produced by SeSAM, we generated arbitrary reference maps with different marker densities and used them to simulate segregation data. Then, for different numbers of individuals or markers (same maps as for Figures 2 and 3), we measured the deviation from colinearity (through Spearman's rank correlation) and the map length ratio between the *framework* (or *total)* map computed from these segregation data and the initial reference map. Finally, we measured the inclusion rate, that is the proportion of markers in the data set that could be included in the map (see Supplementary Tables S2 and S3 in Additional File 2). We see that in F2 or CP populations, the framework maps were always perfectly colinear to the reference map. The *total* maps were also perfectly colinear to their reference map, except for the F2 population with only 50 individuals. The map length ratios were always between 0.88 and 1.03 for the framework map except for very small F2 and CP populations (50 individuals) for which the scaffold could not meet the robustness criteria up to the extremities of the chromosomes and thus dropped some terminal regions. Similar behaviours were observed for the total maps. All markers of the data sets could be included into the total maps except in the case of the F2 population of 50 individuals for which the framework did not cover the whole of the chromosomes as seen before. On the other hand, when looking at the framework maps, we see as expected that their inclusion rate increases with population size, and decreases with the number of markers.

**Sensitivity to data quality** was assessed by simulating data sets with increasing rates of genotyping errors up to extremely high rates (20%). We chose to distribute false data rates uniformly across markers and individuals, although the *simulatePop()* function of SeSAM is able to use Gamma distributions which allow to slide continuously from L-shaped to almost symmetrical distributions. The effect of increasing rates of erroneous data on map quality is shown in Figure 5 for F2 and CP populations without and with activating the error correction option of SeSAM. In both populations, the artefactual inflation of the map due to the genotyping errors is strongly reduced by the error correction algorithm, although high rates of errors cannot be completely corrected, particularly in CP populations. However, in most real data sets, error rates are generally expected to be under 5%, so in such cases, SeSAM correction mostly avoids significant map inflation due to such errors.

**Comparison with other existing softwares.** To assess how the level of map quality achieved by SeSAM compares with that of other mapping softwares currently available, we generated simulated data sets with different numbers of individuals and markers, and ran them with SeSAM as well as with four different programs: IciMapping, ASMap, MapDisto, and TSPmap (see Supplementary Table S4 in Additional File 2). We also tried to use HighMap, but we could not obtain the software from the address mentioned in the paper. All tools tested excepted TSPmap produced high-quality *total* maps, showing high colinearity and similar lengths when compared to the theoretical map used to simulate the segregation data. However, with increasing numbers of markers (>= 10,000), we couldn't get some of the softwares complete the mapping (see Supplementary Table S4 in Additional File 2). Computation times varied a lot between programs, with Lep-map performing much faster than all others, and SeSAM being in the average of the remaining ones. Overall, SeSAM produces maps with at least similar quality as the other softwares tested. Using SeSAM thus allows to

have a fully automatic tool to produce *total* maps with a level of quality similar to most other software currently available, but contrary to those other programs, in addition to producing a total map with all polymorphic markers, SeSAM also automatically selects an optimally large subset of markers to produce a *framework* map statistically robust at any desired LOD threshold.

**Examples with biological data**. To illustrate how SeSAM can perform with real biological data, mapping results obtained from five anonymized experimental data sets from agricultural plant species are presented in Additional File 2 as Supplementary Table S5 and Supplementary Figures S1 to S5. The corresponding anonymized data sets are available as Additional File 5. The number of markers included in the *framework* map was always lower than the total number of polymorphic markers, because no more markers could be included without losing the order robustness at the given default LOD threshold (3.0). In BC_ano, RIL1_ano, and CP_ano data sets, which have small population sizes, the frameworks include less markers than in F2_ano and RIL2_ano, which have larger populations (see Supplementary Table S5 in Additional File 2). This is expected because there are more informative crossovers contributing to the order information in the latter. Moreover, the backcross-derived BC_ano data contain less crossovers (only one effective meiosis) than the other populations, which contributes to the fact that relatively few markers could be incorporated to the BC_ano framework map. Finally, with similar population sizes, the CP_ano map included less markers in its framework map than RIL1_ano. This is partly due to the fact that not all 2-point marker configurations are informative in CP populations (e.g. there is no linkage information between one male pseudo-backcross marker and one female pseudo-backcross marker). As expected, the number of markers in the framework map is thus commensurate with population size and population type since this ensures statistically

supported marker orders. Considering now the *total* maps obtained after placement, they include almost all polymorphic markers for all data sets, the few non-mapped markers being unlinked to any linkage group, or linked to several linkage groups with similar LODs.

To visualy assess the quality of maps produced, SeSAM generates heat maps of pairwise 2-point LOD matrices. If the quality of the map is good, such heat maps should display a smooth decreasing gradient when going away from the diagonal (see left panels of Supplementary Figures S1 to S5 in Additional File 2). Another useful graph generated by SeSAM to assess map quality is the Marey map, which represents the genetic positions *vs* the physical positions of the markers. The derivative of the Marey map curve gives the local values of recombination rate along the chromosomes (called recombination landscape). If the quality of both physical and genetic maps is high, Marey maps are supposed to be smooth and always increasing (see right panels of Supplementary Figures S2 to S5 in Additional File 2). The large flat regions observed with BC_ano, F2_ano, RIL2_ano, and CP_ano typically correspond to the low peri-centromeric recombination rates. For the BC_ano data set however (see Supplementary Figures S1 in Additional File 2), the Marey map is non-monotonic. Since the 2-point linkage matrix indicates a high-quality genetic mapping, the quality of the physical map may be questionable here. Elsewhere, the case of the RIL1_ano data set illustrates the possibilty of using a previously existing genetic map to guide the choice of the seed markers to initiate the seriation process, when no physical map is available. In such cases, the ‘phyMap.txt’ file actually contains a genetic map, so the ‘Marey map’ obtained has an almost constant slope, but it may also be used to compare recombination landscapes between different crosses.

Finally, using the same general algorithm as SeSAM, but with earlier generations of codes, we already produced genetic maps used in several published studies on Maize [6, 34, 37–42], Pea [43–45], and Faba bean [46].

## Conclusions

Compared to existing mapping software, SeSAM is to our knowledge the only one to carry out a completely automatic bin-mapping procedure producing first a mid-density *framework* map from an optimized subset of markers which allow the order to be statistically supported at the desired statistical threshold, and then a high-density *total* map including nearly all polymorphic markers, but preserving the global structure and length of the framework map. SeSAM is freely available to all users, including the source code, and is compatible with Linux, macOS, or Windows platforms.

## Availabillity and requirements

Project name: SeSAM

Project home page: https://forgemia.inra.fr/gqe-acep/sesam

Operating systems: GNU Linux, macOS (>=10.13), Windows10

Programming language: R, C++

Other requirements: the following C++ libraries are required when compiling the package from source: gmp, boost-dev, boost-math (>= 1.56).

License: GNU GPL v3

Any restrictions to use by non-academics: none

## List of abbreviations

CP: Cross-pollinated

BC: Back-cross

DH: Doubled-haploid

EM: Expectation-maximization

IRIL: Intermated recombinant inbred line

LOD: Logarithm of odds

RIL: Recombinant inbred line

SNP: Single-nucleotide polymorphism

## Declarations

**Ethics approval and consent to participate** Not applicable

**Consent for publication** Not applicable

**Availability of data and materials**

All data and codes generated or analysed during this study are included in this published article and its supplementary information files.

**Competing interests** The authors declare that they have no competing interests

**Funding**

This work has received funding from the French Government managed by the Research National Agency (ANR) under the “Investment for the Future” program Amaizing (project ANR-10-BTBR-03), from a French State grant (LabEx Saclay Plant Sciences-SPS, ANR-10-LABX-0040-SPS), managed by the French National Research Agency under an “Investments for the Future” program (ANR-11-IDEX-0003-02), and from MARS-WRIGLEY which co-funded the

salary of AV and WR. The funding bodies did not play any role in the design of the study, collection, analysis, and interpretation of the data, or writing the manuscript.

**Authors' contributions**

MF and OCM designed the general algorithms, AV, WR, NC, NG developed the R mapping functions, ET developed the R data set simulation functions, DL, SJ, FG developed the C++ SpellMapTools functions, MF wrote the manuscript, and all authors approved the manuscript.

**Acknowledgements**

The authors thank E. Bauer, G. Boutet, E. Jenczewski, C. Knaak, S. Mezmouk, N.Tayeh, and V. Geffroy for helpful discussions on the mapping approach and for providing data sets to test, calibrate, and improve the software.

## References

1. Morgan TH. Chromosomes and Heredity. The American Naturalist. 1910;44:449–96.

2. Darvasi A, Weinreb A, Minke V, Weller JI, Soller M. Detecting marker-QTL linkage and estimating QTL gene effect and map location using a saturated genetic map. Genetics. 1993;134:943–51.

3. Causse MA, Fulton TM, Cho YG, Ahn SN, Chunwongse J, Wu K, et al. Saturated molecular map of the rice genome based on an interspecific backcross population. Genetics. 1994;138:1251–74.

4. Doniskeller H. A genetic linkage map of the human genome. Cell. 1987;51:319–37.

5. Deokar AA, Ramsay L, Sharpe AG, Diapari M, Sindhu A, Bett K, et al. Genome wide SNP identification in chickpea for use in development of a high density genetic map and improvement of chickpea reference genome assembly. BMC Genomics. 2014;15:708.

6. Ganal MW, Durstewitz G, Polley A, Bérard A, Buckler ES, Charcosset A, et al. A Large Maize (Zea mays L.) SNP Genotyping Array: Development and Germplasm Genotyping, and Genetic Mapping to Compare with the B73 Reference Genome. PLoS ONE. 2011;6:e28334.

7. Zheng C, Boer MP, Eeuwijk FA van. Construction of Genetic Linkage Maps in Multiparental Populations. Genetics. 2019;212:1031–44.

8. Ritter E, Gebhardt C, Salamini F. Estimation of recombination frequencies and construction of RFLP linkage maps in plants from crosses between heterozygous parents. Genetics. 1990;125:645–54.

9. Grattapaglia D, Sederoff R. Genetic linkage maps of Eucalyptus grandis and Eucalyptus urophylla using a pseudo-testcross: mapping strategy and RAPD markers. Genetics. 1994;137:1121–37.

10. Burr B, Burr FA, Thompson KH, Albertson MC, Stuber CW. Gene Mapping with Recombinant Inbreds in Maize. Genetics. 1988;118:519–26.

11. Beavis W, Lee M, Grant D, Hallauer A, Owens T, Katt M, et al. The influence of random mating on recombination among RFLP loci. Maize Newsl. 1992:52–3.

12. Lee M, Sharopova N, Beavis WD, Grant D, Katt M, Blair D, et al. Expanding the genetic map of maize with the intermated B73 × Mo17 (IBM) population. Plant Molecular Biology. 2002;48:453–61.

13. Cheema J, Dicks J. Computational approaches and software tools for genetic linkage map estimation in plants. Briefings in Bioinformatics. 2009;10:595–608.

14. Jünger M, Reinelt G, Rinaldi G. Chapter 4 The traveling salesman problem. In: Handbooks in Operations Research and Management Science. Elsevier; 1995. p. 225–330.

15. de Givry S, Bouchez M, Chabrier P, Milan D, Schiex T. CarthaGene: multipopulation integrated genetic and radiation hybrid mapping. Bioinformatics. 2004;21:1703–4.

16. Iwata H, Ninomiya S. AntMap: Constructing Genetic Linkage Maps Using an Ant Colony Optimization Algorithm. Breeding Science. 2006;56:371–7.

17. Wu Y, Bhat PR, Close TJ, Lonardi S. Efficient and Accurate Construction of Genetic Linkage Maps from the Minimum Spanning Tree of a Graph. PLoS Genetics. 2008;4:e1000212.

18. Monroe JG, Allen ZA, Tanger P, Mullen JL, Lovell JT, Moyers BT, et al. TSPmap, a tool making use of traveling salesperson problem solvers in the efficient and accurate construction of high-density genetic linkage maps. BioData Mining. 2017;10:38.

19. Baird NA, Etter PD, Atwood TS, Currey MC, Shiver AL, Lewis ZA, et al. Rapid SNP Discovery and Genetic Mapping Using Sequenced RAD Markers. PLOS ONE. 2008;3:e3376.

20. Elshire RJ, Glaubitz JC, Sun Q, Poland JA, Kawamoto K, Buckler ES, et al. A Robust, Simple Genotyping-by-Sequencing (GBS) Approach for High Diversity Species. PLoS ONE. 2011;6:e19379.

21. Sun X, Liu D, Zhang X, Li W, Liu H, Hong W, et al. SLAF-seq: An Efficient Method of Large-Scale De Novo SNP Discovery and Genotyping Using High-Throughput Sequencing. PLOS ONE. 2013;8:e58700.

22. Dempster AP, Laird NM, Rubin DB. Maximum Likelihood from Incomplete Data Via the EM Algorithm. Journal of the Royal Statistical Society: Series B (Methodological). 1977;39:1–22.

23. Rastas P. Lep-MAP3: robust linkage mapping even for low-coverage whole genome sequencing data. Bioinformatics. 2017;33:3726–32.

24. Castiglioni P, Pozzi C, Heun M, Terzi V, Müller KJ, Rohde W, et al. An AFLP-Based Procedure for the Efficient Mapping of Mutations and DNA Probes in Barley. Genetics. 1998;149:2039–56.

25. Lincoln SE, Lander ES. Systematic detection of errors in genetic linkage data. Genomics. 1992;14:604–10.

26. Douglas JA, Boehnke M, Lange K. A Multipoint Method for Detecting Genotyping Errors and Mutations in Sibling-Pair Linkage Data. The American Journal of Human Genetics. 2000;66:1287–97.

27. Os HV, Stam P, Visser RGF, Eck HJV. RECORD: a novel method for ordering loci on a genetic linkage map. Theor Appl Genet. 2005;112:30–40.

28. Cartwright DA, Troggio M, Velasco R, Gutin A. Genetic Mapping in the Presence of Genotyping Errors. Genetics. 2007;176:2521–7.

29. Ronin YI, Mester DI, Minkov DG, Akhunov E, Korol AB. Building Ultra-high Density Linkage Maps Based on Efficient Filtering of Trustable Markers. Genetics. 2017;206:1285-1295.

30. Balcárková B, Frenkel Z, Škopová M, Abrouk M, Kumar A, Chao S, et al. A High Resolution Radiation Hybrid Map of Wheat Chromosome 4A. Frontiers in Plant Science. 2017;7.

31. Sesiz U, Özkan H. A new genetic linkage map in einkorn wheat (Triticum monococcum) detects two major QTLs for heading date in chromosome 2A and 5A, probably corresponding to the photoperiod and vernalization genes. Plant Breeding. 2022;141:12–25.

32. Gardiner JM, Coe EH, Melia-Hancock S, Hoisington DA, Chao S. Development of a Core RFLP Map in Maize Using an Immortalized F(2) Population. Genetics. 1993;134:917–30.

33. Albini G, Falque M, Joets J. ActionMap: a web-based software that automates loci assignments to framework maps. Nucl Acids Res. 2003;31:3815–8.

34. Falque M, Decousset L, Dervins D, Jacob A-M, Joets J, Martinant J-P, et al. Linkage Mapping of 1454 New Maize Candidate Gene Loci. Genetics. 2005;170:1957–66.

35. Buetow KH, Chakravarti A. Multipoint gene mapping using seriation. I. General methods. Am J Hum Genet. 1987;41:180–188.

36. Meng L, Li H, Zhang L, Wang J. QTL IciMapping: Integrated software for genetic linkage map construction and quantitative trait locus mapping in biparental populations. The Crop Journal. 2015;3:269–83.

37. Chardon F, Virlon B, Moreau L, Falque M, Joets J, Decousset L, et al. Genetic Architecture of Flowering Time in Maize As Inferred From Quantitative Trait Loci Meta-analysis and Synteny Conservation With the Rice Genome. Genetics. 2004;168:2169–85.

38. Massonneau A, Houba-Hérin N, Pethe C, Madzak C, Falque M, Mercy M, et al. Maize cytokinin oxidase genes: differential expression and cloning of two new cDNAs. Journal of Experimental Botany. 2004;55:2549–57.

39. Bauer E, Falque M, Walter H, Bauland C, Camisan C, Campo L, et al. Intraspecific variation of recombination rate in maize. Genome Biology. 2013;14:R103.

40. Giraud H, Bauland C, Falque M, Madur D, Combes V, Jamin P, et al. Reciprocal Genetics: Identifying QTL for General and Specific Combining Abilities in Hybrids Between Multiparental Populations from Two Maize (Zea mays L.) Heterotic Groups. Genetics. 2017;207:1167–80.

41. Giraud H, Bauland C, Falque M, Madur D, Combes V, Jamin P, et al. Linkage Analysis and Association Mapping QTL Detection Models for Hybrids Between Multiparental Populations from Two Heterotic Groups: Application to Biomass Production in Maize (Zea mays L.). G3: Genes, Genomes, Genetics. 2017;7:3649-3657.

42. Virlouvet L, El Hage F, Griveau Y, Jacquemot M-P, Gineau E, Baldy A, et al. Water Deficit-Responsive QTLs for Cell Wall Degradability and Composition in Maize at Silage Stage. Front Plant Sci. 2019;10.

43. Tayeh N, Aluome C, Falque M, Jacquin F, Klein A, Chauveau A, et al. Development of two major resources for pea genomics: the GenoPea 13.2K SNP Array and a high-density, high-resolution consensus genetic map. Plant J. 2015;84:1257–73.

44. Boutet G, Alves Carvalho S, Falque M, Peterlongo P, Lhuillier E, Bouchez O, et al. SNP discovery and genetic mapping using genotyping by sequencing of whole genome genomic DNA from a pea RIL population. BMC Genomics. 2016;17:121.

45. Kreplak J, Madoui M-A, Cápal P, Novák P, Labadie K, Aubert G, et al. A reference genome for pea provides insight into legume genome evolution. Nat Genet. 2019;51:1411–22.

46. Carrillo-Perdomo E, Vidal A, Kreplak J, Duborjal H, Leveugle M, Duarte J, et al. Development of new genetic resources for faba bean ( Vicia faba L.) breeding through the discovery of gene-based SNP markers and the construction of a high-density consensus map. Sci Rep. 2020;10:1–14.

## Figures legends

**Figure 1.** General algorithm of automatic map construction in SeSAM. **A**: Construction of the *scaffold* map by seriation. **B**: Densification of the scaffold to produce the high-robustness *framework* map. **C**: Placement of all polymorphic markers to produce the high-density *total* map.

**Figure 2.** SeSAM computation time for automatic *total* map construction, as a function of the density of markers in a F2 (black circles and line) and in a CP (red triangles and line) population of 100 individuals. Data were simulated using the SeSAM function *simulatePop()* for two chromosomes (100 and 200 cM) with markers regularly spaced. Lines were obtained from linear regression y=a.x+b (a=0.007, b=1.9 for F2 and a=0.035, b=3.4 for CP).

**Figure 3.** SeSAM computation time for automatic *total* map construction, as a function of the number of individuals in a F2 (black circles and line) and in a CP (red triangles and line) population. Data were simulated using the SeSAM function *simulatePop()* for two chromosomes (100 and 200 cM) with markers regularly spaced at a density of 1 marker/cM. For F2, the line was obtained from linear regression y=a.x+b (a=0.022, b=0.99). For CP, the line was obtained from non-linear regression $y=a.x^2+b.x+c$ (a=0, b=0.066, c=0.99).

**Figure 4.** SeSAM computation times for automatic *total* map construction for different types of mapping populations of 200 individuals. Data were simulated using the SeSAM function *simulatePop()* for two chromosomes (100 and 200 cM) with markers regularly spaced at a density of 1 marker/cM. Error bars indicate 95% confidence intervals based on nine independent replicates with different seeds for the random number generator used to simulate the data.

**Figure 5.** Map length ratio between the *framework* map and the simulated reference map after SeSAM automatic map construction, as a function of the percentage of genotyping errors in a F2 (left panel) or CP (right panel) population of 200 individuals, without and with activating the genotyping error correction option of SeSAM (black circles and red triangles respectively). Data were simulated using the SeSAM function *simulatePop()* for two

chromosomes (100 and 200 cM) with markers regularly spaced at a density of 1 marker/cM, with increasing proportions of genotyping errors uniformly distributed along chromosomes. Lines were obtained by non-linear regression y=a*sqrt(x)+b*x+c (values of (a,b,c) without and with genotyping error correction, leading to respectively (1.09, -0.06, 0.64) and (0.07, 0.025, 0.89) for F2, and respectively (0.93, -0.049, 0.64) and (0.20, 0.022, 0.84) for CP). Error bars indicate 95% confidence intervals based on five independent replicates with different seeds for the random number generator used to simulate the data. Dotted lines indicate the theoretical outcome of a perfect genotyping error correction (y=1).

## Additional Files

### Additional file 1

Additional_file_1.zip

Compressed archive (.zip)

SeSAM 1.0.1 Packages for Linux, Windows, and macOS platforms.

The archive contains the following folders and files:

| | |
|---|---|
| README | Instructions to install SesSAM on different platforms. |
| GNU_Linux/sesam-1.0.1.tar.gz | R package including all R and C++ source codes. |

Use this archive to install SeSAM on GNU Linux.

| | |
|---|---|
| Windows10_R4.1.x/SeSAM_1.0.1.zip | R package, binary build for Windows10 with R 4.1.x |
| Windows10_R4.2.0/SeSAM_1.0.1.zip | R package, binary build for Windows10 with R 4.2.0 |
| Windows10_R4.2.1/SeSAM_1.0.1.zip | R package, binary build for Windows10 with R 4.2.1 |
| macOS/SeSAM_1.0.1.macos.tgz | R package, binary build for macOS with R 4.2.x |

### Additional file 2

Additional_file_2.pdf

Printable document format (.pdf)

Supplementary Data.

The file contains the following Supplementary Text, Tables and Figures:

*Supplementary Text S1*: Description of the main functions used in SeSAM

*Supplementary Table S1*: Non-exhaustive list of software tools available for linkage mapping.

*Supplementary Table S2:* Quality assessment of the maps produced using SeSAM in the benchmarks presented in Figure 2.

*Supplementary Table S3:* Quality assessment of the maps produced using SeSAM in the benchmarks presented in Figure 3.

*Supplementary Table S4:* Comparison of the quality of maps produced by SeSAM and four of the other mapping softwares listed in Supplementary Table S1.

*Supplementary Table S5:* Summary of maps obtained with SeSAM using experimental data sets from three agricultural plant species.

*Supplementary Figure S1:* Map quality assessment graphs for SeSAM when using the BC_ano experimental data set (BC1 population).

*Supplementary Figure S2:* Map quality assessment graphs for SeSAM when using the F2_ano experimental data set (F2 population).

*Supplementary Figure S3:* Map quality assessment graphs for SeSAM when using the RIL1_ano experimental data set (RIL population).

*Supplementary Figure S4:* Map quality assessment graphs for SeSAM when using the RIL2_ano experimental data set (RIL population).

*Supplementary Figure S5:* Map quality assessment graphs for SeSAM when using the CP_ano experimental data set (CP population from heterozygous outbred parents).

**Additional file 3**

Additional_file_3.zip

Compressed archive (.zip)

The archive contains two files:

| | |
|---|---|
| SeSAM_user-manual.pdf | Reference User Manual in printable document format (.pdf) |
| SeSAM_Quick-Start_tutorial.R | R script to quickly learn and test most SeSAM functions |

**Additional file 4**

Additional_file_4.zip

Compressed archive (.zip)

SeSAM Benchmarking Scripts

The archive contains the data and scripts used to carry out the benchmarking of SeSAM based on simulated data sets and to produce Figures 2, 3, 4, 5, and Supplementary Tables S2 and S3.

**Additional file 5**

Additional_file_5.zip

Compressed archive (.zip)

Experimental Data Sets

The archive contains the following anonymized experimental mapping data sets from agricultural plants, used to produce the maps presented in Supplementary Table S5 and Supplementary Figures S1 to S5:

BC_ano_segData.raw

BC_ano_phyMap.txt

F2_ano_segData.raw

F2_ano_phyMap.txt

RIL1_ano_segData.raw

RIL1_ano_phyMap.txt

RIL2_ano_segData.raw

RIL2_ano_phyMap.txt

CP_ano_segData.gen

CP_ano_phyMap.txt