Nucleic Acids Research, 2025, 1–4
https://doi.org/10.1093/nar/gkaf332
Web Server issue

MSTmap Online: enhanced usability, visualization, and accessibility

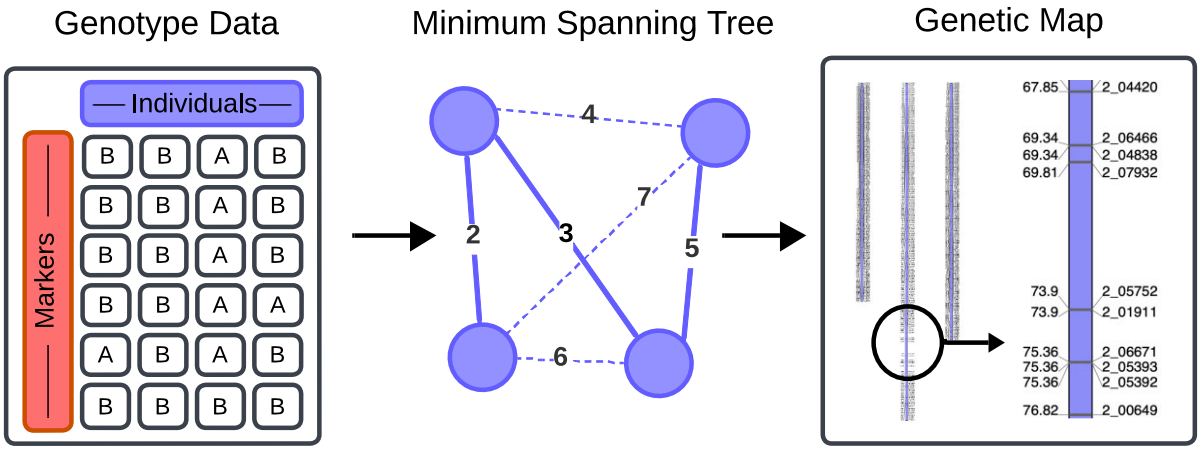
Amirsadra Mohseni and Stefano Lonardi*

Computer Science and Engineering, University of California, Riverside, CA 92521, United States
*To whom correspondence should be addressed. Email: stelo@cs.ucr.edu

Abstract

Genetic linkage maps are an essential tool in population genetics and plant breeding research, yet user-friendly online tools for constructing and visualizing them remain scarce. MSTmap Online addresses this gap by providing a modern, accessible platform for generating high-quality genetic linkage maps from genotypic data. The web server quickly computes linkage groups using the MSTmap algorithm and generates detailed output files, including publication-ready PDF visualizations of linkage groups. The server supports bookmarking and asynchronous processing, allowing users to revisit their results at a later time. A companion Python library for MSTmap Online enables seamless integration into custom analysis pipelines. MSTmap Online is free and open to all users with no login requirement at https://mstmap.org. The companion Python library is available at https://pypi.org/project/mstmap/.

Graphical abstract



Introduction

Genetic linkage mapping has been a cornerstone of genetics research since the early twentieth century, enabling researchers to understand the recombinational behavior and organization of chromosomes. With the introduction of DNA-based genetic markers based on restriction fragment length polymorphism, simple sequence repeats, diversity arrays technology, and amplified fragment length polymorphism, among others, genetic maps have become increasingly dense and accurate, facilitating biological studies such as marker-assisted breeding and map-based cloning. As marker densities continue to rise with advancements in genotyping technologies, the computational challenge of constructing linkage maps with thousands or even millions of markers has grown substantially. While several published tools are available for constructing genetic maps, very few are accessible as online tools, and they require some knowledge of either of command-line interfaces or programming (see Supplementary Table S1 for a list). Thus, there is a need for accessible, scalable, and user-friendly platforms to support researchers handling increasingly complex datasets.

MSTmap is a time- and memory-efficient algorithm capable of constructing accurate genetic linkage maps, even in the presence of genotyping errors or missing genotyping calls. It can handle various inbred biparental mapping populations, including doubled haploid (DH), haploid (Hap), recombinant inbred line (RIL), and advanced RIL, for up to ∼10 000 markers and ∼1000 individuals. Prior to MSTmap, genetic maps were constructed by solving variants of the traveling

Received: January 31, 2025. **Revised:** March 12, 2025. **Editorial Decision:** April 8, 2025. **Accepted:** April 14, 2025

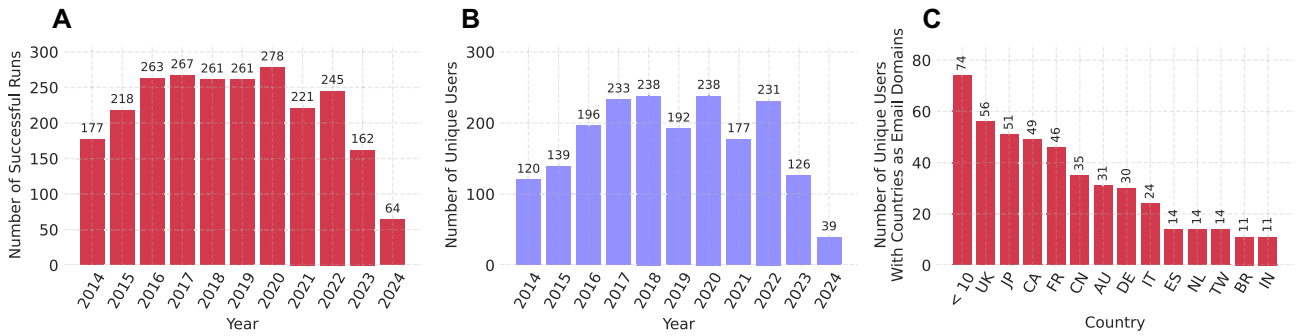


Figure 1. (**A**) Total number of successful runs completed on the original MSTmap web server. (**B**) Annual count of unique users utilizing the original MSTmap web server at least once. (**C**) Geographical distribution of MSTmap users based on the country code in their email addresses, when available. *Note*: Email addresses from countries appearing fewer than 10 times are grouped under the "<10" bar on the *x*-axis.

salesman path (TSP) problem on a graph whose nodes were the genetic markers and the weights on the edges represented genetic distances. We recall that a TSP for a graph G is the shortest path (i.e. a path of minimum total weight) that visits every marker/vertex of G exactly once. It is known that finding a TSP is computationally NP-hard [1]. In [2], the authors of MSTmap proved that under some general assumptions on the function that assigns weights to the edges of the graph, the minimum weight TSP of G corresponds to the correct order of genetic markers. Furthermore, they proved that when the minimum spanning tree (MST) of G is unique, the minimum weight TSP of G (and thus, the correct order of the markers) can be computed by a simple MST algorithm, such as Prim's algorithm [3, 4].

Since its publication in 2008, MSTmap has been used for many genetic mapping projects in plants, as reflected by over 600 citations (source: Google Scholar) to the paper published in *PLoS Genetics* [2]. For instance, MSTmap was used to generate high-density genetic linkage maps for cotton [5], wheat [6–8], sugarcane [9], cowpea [10, 11], rice [12], melons [13], and sunflowers [14], among others. The original MSTmap web server was initially deployed in 2014 to offer its functionalities to the broadest possible plant community. Over the span of 11 years, the original web server had over 55 000 accesses and was used by about 2000 unique users from countries all around the world (Fig. 1).

Despite its popularity, the original MSTmap web server (based on HTML4 and CGI) had a series of limitations that hampered its usability. These included the inability to visualize the genetic maps, limited data confidentiality, poor scalability for large datasets, and lack of multi-user support. These limitations motivated the development of MSTmap Online.

Materials and methods

The web server stack

The new MSTmap web server is built using the Python Flask framework and can handle up to ∼10 000 genetic markers and ∼1000 individuals. The backend of this framework communicates with the core MSTmap algorithm through a Cython-based Python interface that wraps the original C++ implementation. This integration allows the efficient processing of input genotype data and a quick generation of the linkage groups, their size and bin counts, marker positions, and detailed map visualizations. A PDF is generated to visualize linkage groups across chromosomes using a modified version of the Biopython Graphics library [15]. Users can also download the results as plain text through a dedicated download page, or they can provide an optional email address that will be used to alert them when the processing is complete.

The web server workflow is simple and intuitive. Users upload genotype files as a text file in a tab-delimited format, customize algorithm parameters if needed, and receive processed results. By default, algorithm parameters are preset, but advanced users can modify thresholds, population types, and other settings. Detailed documentation for these parameters is provided on the Help page. Fig. 2A illustrates the steps required to upload the input, modify the parameters, and receive a processed output. Fig. 2B shows an example of a visualization of the genetic map, which is the order of the input markers and their distances in centimorgans. This visual representation of the map is made available as a downloadable PDF, enabling users to inspect, share, or include it in their own publications.

Internally, the server uses Celery, an asynchronous task queue system, to manage computationally intensive tasks such as the processing of large data input and the creation of the visual representation of the map. Celery allows the server to offload tasks to background workers, ensuring that the web interface remains responsive. When a genotypic dataset is uploaded, the task is queued and executed in the background, enabling users to continue interacting with the application without waiting for the computation to finish.

Companion Python library

As part of the development of MSTmap Online, we integrated the MSTmap algorithm with Python through the development of a Cython wrapper. This wrapper bridges the original C++ implementation with a Python library, enabling seamless communication while leveraging the efficiency of C++ and the flexibility of Python. A similar integration (called ASMap) has been available for the R language for several years [16]. Our MSTmap Python integration is utilized in the web server implementation and provided as a stand-alone library for custom workflows.

When using MSTmap within a larger Python code, we made sure that the parameters of MSTmap could be either embedded in the input file or set through Python function calls. Users can provide the input file path to the Python interface or use default parameters for quick configuration. Alternatively, parameters can be individually specified or overridden using Python function calls, granting fine-grained control over al-

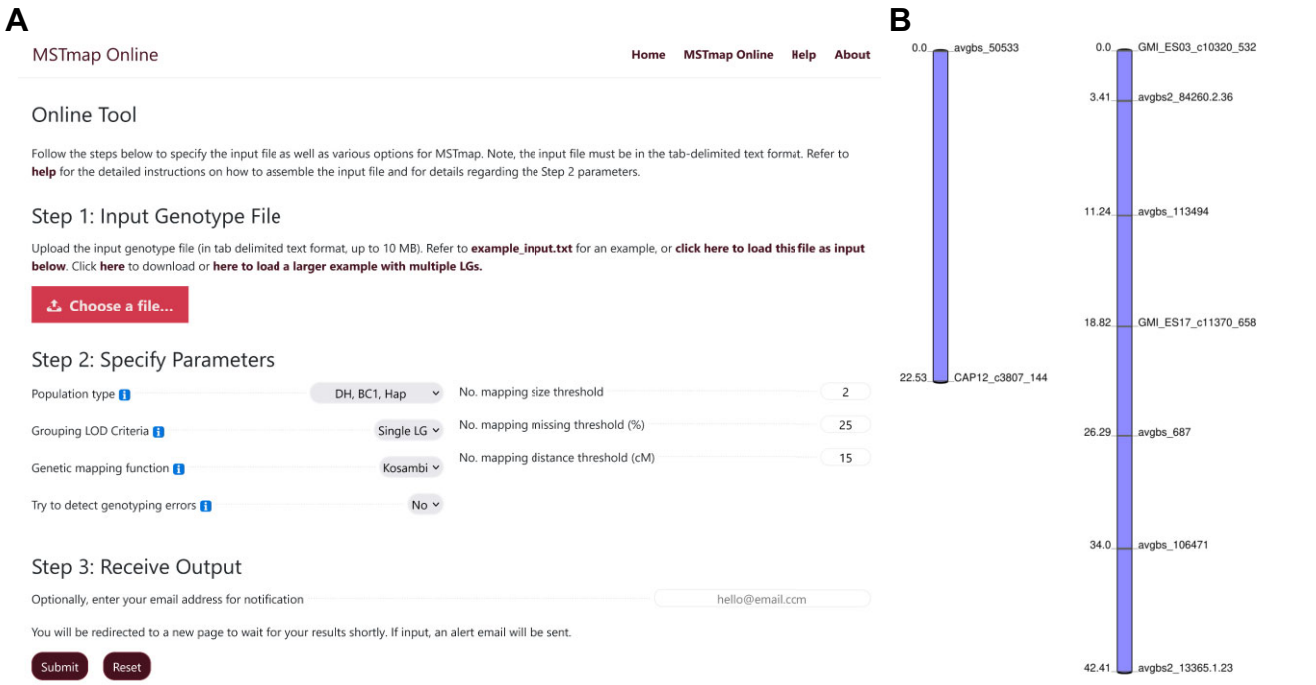


Figure 2. (**A**) Step 1: Users begin by uploading a genotypic dataset, which must include both locus names and individual genotype data. Two example files are provided as a reference to demonstrate the expected format. For convenience, users have the option to directly load the example files as input without needing to download and upload them manually. Step 2: Users can customize the algorithm by specifying relevant parameters and the population type. Details for these parameters are provided via both tool-tips and the Help page. Step 3: Users have the option to include an email, which is used to notify them when the processing is complete, including a unique link to retrieve their results. (**B**) A genetic map is generated as a PDF file based on the processed output.

gorithm behavior. This flexibility enables researchers to create scripts for automated workflows, dynamically adjust parameters based on intermediate results, and integrate MSTmap into larger data analysis pipelines.

As for the web server, the input file to MSTmap is a tab-delimited text file where each cell in the table refers to the genotype state of a particular mapping line on a particular marker locus. The output includes the number of linkage groups, the size and number of bins in each linkage group, and the order and the distance of each marker in each linkage group. The optimal order of the markers is rapidly computed by utilizing a minimal spanning tree of the graph obtained from the genotypic data. The documentation for the Python library is provided in the supplementary material, along with a step-by-step use-case example of creating a mapping for over 14 000 markers.

Data protection

The MSTmap Online web server is designed with robust security measures to protect user privacy and ensure data confidentiality. The website uses HTTPS with an SSL/TLS certificate. IP addresses are not logged, and cookies are not used to track user activity. Uploaded files are validated for format and content before processing, with invalid or malformed files prompting an immediate error message on the web page.

Valid genotype files are temporarily stored in a secure "upload" folder with randomized names generated using 128-bit UUIDs. During processing, each user is assigned a unique UUID URL, enabling them to bookmark and return to their results later. The UUID acts essentially as an "unguessable password" embedded in the URL, and the web server does not inform the user of the invalidity of the URL when they visit a page that was not assigned to them. Once processing is complete, results are stored in a dedicated "download" folder. Users may optionally provide an email address to receive a notification with a retrieval URL that expires after 24 h. Uploaded files are deleted immediately after processing, and results are automatically purged after either (i) they are downloaded or (ii) 24 h have elapsed, whichever comes first. These measures ensure maximum data security and privacy.

Parameter selection

As mentioned earlier, MSTmap can handle genotypic data from various inbred biparental mapping populations. Users are expected to select the correct parameters based on the characteristics of the data provided in input, including the population type (DH, BC1, Hap, RILs), marker count, and data quality. For instance, users are recommended to turn off the correction of genotypic errors unless they know that their data are of low quality. Selecting the correct logarithm of the odds (LOD) score can require some trial and error. If the number of chromosomes is known beforehand, users can iterate through LOD scores until the highest LOD score yielding the expected number of chromosomes is identified.

Discussion

Here, we presented a new MSTmap Online web server and a new MSTmap Python library, offering significant advancements in both usability and functionality. Compared to the original MSTmap web server, the new server introduces a modern user interface, enhanced output visualization, a robust

asynchronous queue system for processing large datasets, and the ability for users to bookmark results and return to them later. These updates address many of the limitations of the previous version and expand the tool's applicability to a broader audience, including users with more complex or large-scale data processing needs.

The updated server is highly efficient, secured with SSL certification, and built with a focus on privacy by (i) avoiding the use of cookies or tracking and (ii) not saving user data for extended periods. It remains free to use for both academic and commercial users, offering a streamlined, user-friendly experience while ensuring robust functionality. With these improvements, we are confident that the new MSTmap Online web server will become a valuable resource for plant geneticists, botanists, and the broader scientific community.

Acknowledgements

Author contributions: Amirsadra Mohseni: Software, Visualization, Writing – original draft. Stefano Lonardi: Conceptualization, Funding acquisition, Supervision, Writing – review & editing.

Supplementary data

Supplementary data is available at NAR online.

Conflict of interest

None declared.

Funding

The initial development of MSTmap was supported by a US National Science Foundation grant [DBI-0321756]. This work was supported in part by the US National Science Foundation [CBET-2225878, IIS-244456]. Funding to pay the Open Access publication charges for this article was provided by US National Science Foundation [CBET-2225878] and US National Institutes of Health [1R01AI169543-01].

Data availability

The web server is available at https://mstmap.org. It is completely free and open, and does not require registration. The source code and documentation for the Python library are freely available at https://github.com/ucrbioinfo/MSTmap-Python and https://doi.org/10.5281/zenodo.15192381, and the package is available on PyPI: https://pypi.org/project/mstmap/.

References

1. Karp RM. Reducibility among combinatorial problems. In: Jünger M, Liebling TM, Naddef D *et al.* . (eds.), *50 Years of Integer Programming 1958–2008*. Berlin: Springer, 2010, 219–41. https://doi.org/10.1007/978-3-540-68279-0_8
2. Wu Y, Bhat PR, Close TJ *et al.* Efficient and accurate construction of genetic linkage maps from the minimum spanning tree of a graph. *PLoS Genet* 2008;**4**:e1000212. https://doi.org/10.1371/journal.pgen.1000212
3. Jarník V. O jistém problému minimálním [About a certain minimal problem (in Czech)]. *Práca Moravské Prírodovedecké Spolecnosti* 1930;**6**:57–63.
4. Prim RC. Shortest connection networks and some generalizations. *Bell System Tech J* 1957;**36**:1389–1401. https://doi.org/10.1002/j.1538-7305.1957.tb01515.x
5. Hu Y, Chen J, Fang L *et al.* *Gossypium barbadense* and *Gossypium hirsutum* genomes provide insights into the origin and evolution of allotetraploid cotton. *Nat Genet* 2019;**51**:739–48. https://doi.org/10.1038/s41588-019-0371-5
6. Wang S, Wong D, Forrest K *et al.* Characterization of polyploid wheat genomic diversity using a high-density 90,000 single nucleotide polymorphism array. *Plant Biotechnol J* 2014;**12**:787–96. https://doi.org/10.1111/pbi.12183
7. Juliana P, Poland J, Huerta-Espino J *et al.* Improving grain yield, stress resilience and quality of bread wheat using large-scale genomics. *Nat Genet* 2019;**51**:1530–9. https://doi.org/10.1038/s41588-019-0496-6
8. Ling HQ, Ma B, Shi X *et al.* Genome sequence of the progenitor of wheat A subgenome *Triticum urartu*. *Nature* 2018;**557**:424–8. https://doi.org/10.1038/s41586-018-0108-0
9. Garsmeur O, Droc G, Antonise R *et al.* A mosaic monoploid reference sequence for the highly complex genome of sugarcane. *Nat Commun* 2018;**9**:2638. https://doi.org/10.1038/s41467-018-05051-5
10. Lonardi S, Muñoz-Amatriaín M, Liang Q *et al.* The genome of cowpea (*Vigna unguiculata* [L.] Walp.). *Plant J* 2019;**98**:767–82. https://doi.org/10.1111/tpj.14349
11. Steinbrenner AD, Muñoz-Amatriaín M, Chaparro AF *et al.* A receptor-like protein mediates plant immune responses to herbivore-associated molecular patterns. *Proc Natl Acad Sci USA* 2020;**117**:31510–8. https://doi.org/10.1073/pnas.2018415117
12. Zhou Y, Yu Z, Chebotarov D *et al.* Pan-genome inversion index reveals evolutionary insights into the subpopulation structure of Asian rice. *Nat Commun* 2023;**14**:1567. https://doi.org/10.1038/s41467-023-37004-y
13. Sun H, Wu S, Zhang G *et al.* Karyotype stability and unbiased fractionation in the paleo-allotetraploid *Cucurbita* genomes. *Mol Plant* 2017;**10**:1293–306. https://doi.org/10.1016/j.molp.2017.09.003
14. Renaut S, Grassa C, Yeaman S *et al.* Genomic islands of divergence are not affected by geography of speciation in sunflowers. *Nat Commun* 2013;**4**:1827. https://doi.org/10.1038/ncomms2833
15. Cock PJ, Antao T, Chang JT *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 2009;**25**:1422. https://doi.org/10.1093/bioinformatics/btp163
16. Taylor J, Butler D. R package ASMap: efficient genetic linkage map construction and diagnosis. *J Stat Softw* 2017;**79**:1–29. https://doi.org/10.18637/jss.v079.i06

Received: January 31, 2025. **Revised:** March 12, 2025. **Editorial Decision:** April 8, 2025. **Accepted:** April 14, 2025