Bioinformatics, 2022, 1–3
https://doi.org/10.1093/bioinformatics/btac505
Advance Access Publication Date: 19 July 2022
Applications Note

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btac505/6646451 by University of Oslo user on 16 August 2022

Sequence analysis

CompAIRR: ultra-fast comparison of adaptive immune receptor repertoires by exact and approximate sequence matching

Torbjørn Rognes [1,2,3,†], Lonneke Scheffer [1,3,†], Victor Greiff [4] and Geir Kjetil Sandve [1,3,*]

[1]Department of Informatics, University of Oslo, 0316 Oslo, Norway, [2]Department of Microbiology, Oslo University Hospital, 0424 Oslo, Norway, [3]Centre of Bioinformatics, University of Oslo, 0316 Oslo, Norway and [4]Department of Immunology, University of Oslo and Oslo University Hospital, 0424 Oslo, Norway

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Alfonso Valencia

Received on December 9, 2021; revised on March 20, 2022; editorial decision on July 2, 2022; accepted on July 18, 2022

Abstract

Motivation: Adaptive immune receptor (AIR) repertoires (AIRRs) record past immune encounters with exquisite specificity. Therefore, identifying identical or similar AIR sequences across individuals is a key step in AIRR analysis for revealing convergent immune response patterns that may be exploited for diagnostics and therapy. Existing methods for quantifying AIRR overlap scale poorly with increasing dataset numbers and sizes. To address this limitation, we developed CompAIRR, which enables ultra-fast computation of AIRR overlap, based on either exact or approximate sequence matching.

Results: CompAIRR improves computational speed 1000-fold relative to the state of the art and uses only one-third of the memory: on the same machine, the exact pairwise AIRR overlap of 10^4 AIRRs with 10^5 sequences is found in ∼17 min, while the fastest alternative tool requires 10 days. CompAIRR has been integrated with the machine learning ecosystem immuneML to speed up commonly used AIRR-based machine learning applications.

Availability and implementation: CompAIRR code and documentation are available at https://github.com/uio-bmi/compairr. Docker images are available at https://hub.docker.com/r/torognes/compairr. The code to replicate the synthetic datasets, scripts for benchmarking and creating figures, and all raw data underlying the figures are available at https://github.com/uio-bmi/compairr-benchmarking.

Contact: geirksa@ifi.uio.no

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Adaptive immune receptor (AIR) repertoires (AIRRs) record past immune encounters. High-throughput sequencing now enables millions of AIR sequences to be determined at a cost that facilitates adaptive immunity-based association studies on large patient cohorts (Emerson *et al.*, 2017; Liu *et al.*, 2019). It has been previously shown that shared immune states give rise to identical or similar AIR sequences across individuals, enabling the use of AIRR-seq for diagnostics and therapeutic research (Arnaout *et al.*, 2021; Greiff *et al.*, 2020). Computation of cross-individual AIRR intersections, i.e. the number of matching AIR sequences across AIRRs, is thus a foundational computational task performed in nearly all AIRR analyses. However, since the number of pairwise AIRR comparisons grows asymptotically quadratically with the number of AIRRs considered, where each pairwise AIRR comparison typically involves millions of individual AIRs, computational efficiency is crucial for performing AIR sequence matching at scale.

We here present CompAIRR, a tool that allows to compute AIRR intersections up to 1000-fold faster than current implementations (Nazarov *et al.*, 2019; Shugay *et al.*, 2015; Weber *et al.*, 2022). In contrast to existing tools, CompAIRR supports both exact and approximate sequence matching between AIRs when determining AIRR overlap. The CompAIRR implementation is available both as a stand-alone command-line tool, and as a component integrated with the machine learning ecosystem immuneML (Pavlović *et al.*, 2021) (from immuneML version 2.1.0 onward) to accelerate the

© The Author(s) 2022. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

computation of AIRR similarity matrices, and to accelerate an AIRR-based immune state classifier (Emerson *et al.*, 2017) that is implemented in the immuneML system (Supplementary Fig. S1).

2 CompAIRR description

CompAIRR is based on a sequence comparison strategy developed for the nucleotide sequence clustering tool Swarm (Mahé *et al.*, 2022). A Bloom filter (Putze *et al.*, 2010) and a hash table are used to quickly look up similar AIR sequences across AIRR sets. For each AIR sequence (nucleotide or amino acid), a 64-bit hash value is generated using a Zobrist hash function (Zobrist, 1970), a form of tabulation hashing that can be computed very efficiently and updated incrementally. When approximate matching is enabled, the hashes of all possible variants of a query sequence (with 1–2 substitutions or indels) are also generated. This search strategy identifies all matching sequences without compromising on accuracy. CompAIRR version 1.7.0 or later also supports a larger number of substitutions by using a simpler all-versus-all algorithm. Matches are optionally restricted by V and J gene. Multi-threading may be enabled to further speed up comparisons (see Fig. 1d). For the comparison of n AIRRs, CompAIRR produces an $n \times n$ matrix where each cell contains the sum of matching AIR frequencies with flexible summary statistics (product, min, max, mean or ratio of the two compared AIR frequencies), or the Morisita-Horn or Jaccard index between AIRRs. Alternatively, CompAIRR can query n AIRRs against m reference AIRs and produce an $n \times m$ sequence presence table. While AIR matching is only supported at the single chain level, two $n \times m$ sequence presence tables for complementary (paired) AIR chains (single-cell data) can easily be merged. For the analysis of a single AIRR, CompAIRR can perform single-linkage clustering of AIRs. CompAIRR can optionally output the list of (approximately) matching AIRs as an AIRR-compliant TSV file, and adheres to the AIRR standard for software tools (Vander Heiden *et al.*, 2018).

3 CompAIRR performance benchmarking

CompAIRR (1.3.1) was benchmarked against VDJtools (1.2.1) (Shugay *et al.*, 2015), immunarch (0.6.5) (Nazarov *et al.*, 2019) and immuneREF (0.5.0) (Weber *et al.*, 2022) by calculating the pairwise AIRR overlap of datasets ranging from 10 to 10^4 AIRRs. Each AIRR consisted of 10^5 amino acid AIR sequences generated using OLGA (1.2.2) (Sethna *et al.*, 2019) with the default human IgH CDR3 model. Figure 1b and c, respectively, shows the running time and maximum RAM usage of each tool. CompAIRR is consistently faster, particularly for large datasets: with 10^4 AIRRs of 10^5 sequences, CompAIRR ran in 17 min while immunarch took 10 days, immuneREF took 23 days and VDJtools failed to complete due to memory constraints. The computational complexity appears to have been reduced from approximately quadratic to almost linear. Furthermore, the maximum RAM usage of CompAIRR is below one-third of that of competing tools. The running time and memory usage as a function of the AIRR size (10^4–10^6 sequences) is shown in Supplementary Figure S2.

In addition, Figure 1d shows how the CompAIRR running time is affected by approximate sequence matching, which is not at all supported by the existing tools. The benefit of multi-threading becomes more apparent when the degree of sequence mismatching is increased, since with exact matching the running time is dominated by disk access (Supplementary Fig. S3).

4 Conclusion

The identification of shared AIRs across AIRRs from different individuals is a core computational task in AIRR analysis. We have here presented CompAIRR, which calculates AIRR overlap up to 1000-fold faster while its peak memory usage is below one third compared to currently available tools. We validated that CompAIRR easily scales to datasets of 10^4 AIRRs of 10^5 sequences each, which surpass the largest available experimental datasets (Liu *et al.*, 2019; Nolan *et al.*, 2020). Furthermore, a novel feature of CompAIRR is efficient

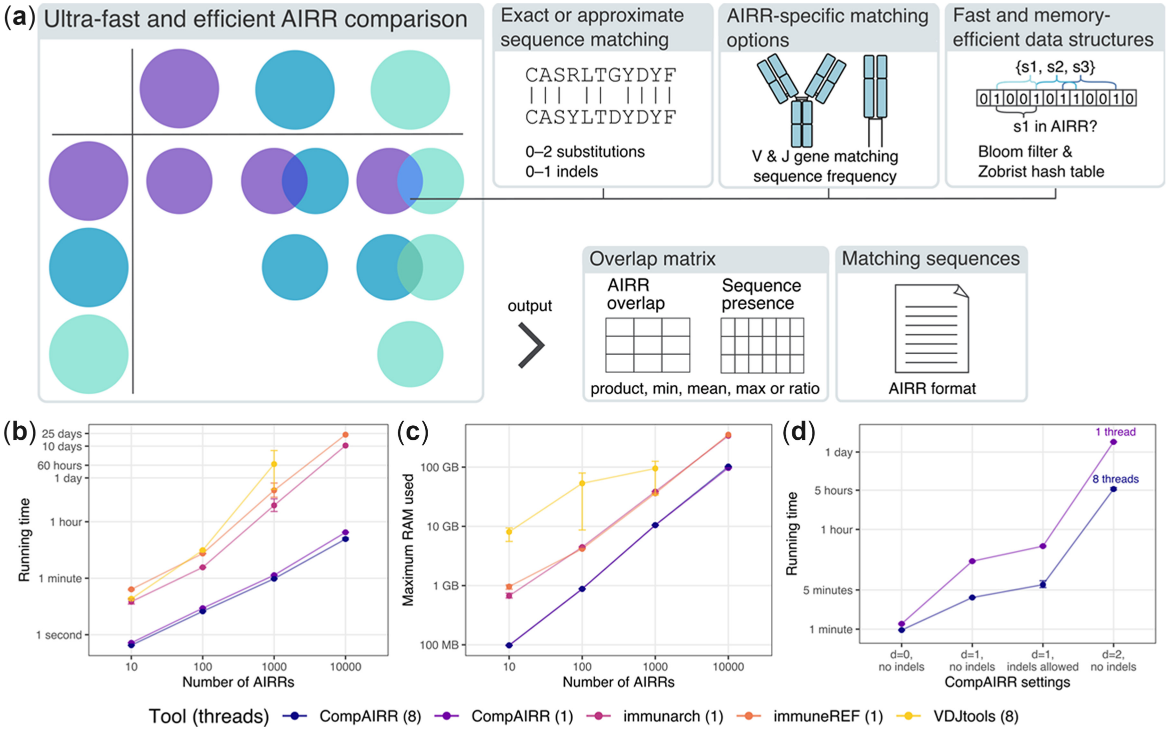

Fig. 1. Overview of CompAIRR features and performance. (**a**) CompAIRR has configurable AIR matching criteria and output formats. (**b**) CompAIRR calculates pairwise AIRR overlap up to 1000-fold faster than currently available tools. (**c**) The maximum RAM usage of CompAIRR is below one-third of the most memory-efficient alternative. (**d**) The CompAIRR running time increases when allowing more AIR sequence mismatches, but multithreading helps reduce this running time. (**b–d**) Data shown are mean with error bars showing min/max values across three replicate runs. For the largest dataset, only CompAIRR was run three times, and VDJtools failed to run due to memory limitations. Unless otherwise specified, datasets consist of 1000 AIRRs containing 10^5 OLGA-generated sequences (Sethna *et al.*, 2019) (default human IgH CDR3 model)

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btac505/6646451 by University of Oslo user on 16 August 2022

identification of *approximately* matching AIR sequences across AIRRs or to reference databases, which may be a biologically meaningful way to increase the number of matches between AIRRs when the exact overlap is low (Supplementary Fig. S4).

Complementary to sequence-level clustering tools ClusTCR (Valkiers *et al.*, 2021) and GIANA (Zhang *et al.*, 2021), or comparison of AIRR subsets (Yohannes *et al.*, 2021), CompAIRR can be used for ultrafast similarity-based comparison of *complete* AIRRs. Due to flexible specification of summary statistics and output, CompAIRR is easily integrated with any tool capable of reading in either (i) a pairwise distance matrix containing cross-AIRR matches, (ii) a matrix showing individual AIR presence in one or more AIRRs or (iii) an AIRR-compliant TSV file containing (approximately) matching AIRs between AIRRs. This allows accelerating a variety of analyses where AIRR comparison is a core computational component, including AIRR similarity (Weber *et al.*, 2022) and clustering (Rempała and Seweryn, 2013; Shugay *et al.*, 2015), phylogenetic clustering (Hoehn *et al.*, 2022), graph analysis (Madi *et al.*, 2017; Miho *et al.*, 2019; Pogorelyy *et al.*, 2019) and immune state classification (Emerson *et al.*, 2017).

Acknowledgements

The authors are grateful to Milena Pavlović for her assistance. They are also grateful for access to computational resources provided by the Norwegian Research and Education Cloud (NREC) and UNINETT Sigma2 (project NN9383K).

Funding

The authors acknowledge generous support by the Leona M. and Harry B. Helmsley Charitable Trust [#2019PG-T1D011 to V.G.], UiO World-Leading Research Community [to V.G.], UiO: LifeScience Convergence Environment Immunolingo [to V.G. and G.K.S.], EU Horizon 2020 iReceptorplus [#825821 to V.G.], a Research Council of Norway FRIPRO project [#300740 to V.G.], a Research Council of Norway IKTPLUSS project [#311341 to V.G. and G.K.S.] and Stiftelsen Kristian Gerhard Jebsen (K.G. Jebsen Coeliac Disease Research Centre) [G.K.S.].

Conflict of Interest: V.G. declares advisory board positions in aiNET GmbH and Enpicom B.V. V.G. is a consultant for Roche/Genentech.

References

Arnaout,R.A. *et al.*; Adaptive Immune Receptor Repertoire Community. (2021) The future of blood testing is the immunome. *Front. Immunol.*, **12**, 626793.

Emerson,R.O. *et al.* (2017) Immunosequencing identifies signatures of cytomegalovirus exposure history and HLA-mediated effects on the T cell repertoire. *Nat. Genet.*, **49**, 659–665.

Greiff,V. *et al.* (2020) Mining adaptive immune receptor repertoires for biological and clinical information using machine learning. *Curr. Opin. Syst. Biol.*, **24**, 109–119.

Hoehn,K.B. *et al.* (2022) Phylogenetic analysis of migration, differentiation, and class switching in B cells. *PLoS Comput. Biol.*, **18**, e1009885.

Liu,X. *et al.* (2019) T cell receptor β repertoires as novel diagnostic markers for systemic lupus erythematosus and rheumatoid arthritis. *Ann. Rheum. Dis.*, **78**, 1070–1078.

Madi,A. *et al.* (2017) T cell receptor repertoires of mice and humans are clustered in similarity networks around conserved public CDR3 sequences. *eLife*, **6**, e22057.

Mahé,F. *et al.* (2022) Swarm v3: towards tera-scale amplicon clustering. *Bioinformatics*, **38**, 267–269.

Miho,E. *et al.* (2019) Large-scale network analysis reveals the sequence space architecture of antibody repertoires. *Nat. Commun.*, **10**, 1321.

Nazarov,V.I. *et al.* (2019) Immunarch: an R package for painless bioinformatics analysis of T-cell and B-cell immune repertoires. *Zenodo*, https://doi.org/10.5281/zenodo.3367200.

Nolan,S. *et al.* (2020) A large-scale database of T-cell receptor beta (TCRβ) sequences and binding associations from natural and synthetic exposure to SARS-CoV-2. *Res Sq*, rs.3.rs-51964.

Pavlović,M. *et al.* (2021) The immuneML ecosystem for machine learning analysis of adaptive immune receptor repertoires. *Nat. Mach. Intell.*, **3**, 936–944.

Pogorelyy,M.V. *et al.* (2019) Detecting T cell receptors involved in immune responses from single repertoire snapshots. *PLoS Biol.*, **17**, e3000314.

Putze,F. *et al.* (2010) Cache-, hash-, and space-efficient bloom filters. *ACM J. Exp. Algorithmics*, **14**, 4:4.4–4:4.18.

Rempała,G.A. and Seweryn,M. (2013) Methods for diversity and overlap analysis in T-cell receptor populations. *J Math Biol*, **67**, 1339–1368.

Sethna,Z. *et al.* (2019) OLGA: fast computation of generation probabilities of B- and T-cell receptor amino acid sequences and motifs. *Bioinformatics*, **35**, 2974–2981.

Shugay,M. *et al.* (2015) VDJtools: unifying post-analysis of T cell receptor repertoires. *PLoS Comput. Biol.*, **11**, e1004503.

Valkiers,S. *et al.* (2021) ClusTCR: a python interface for rapid clustering of large sets of CDR3 sequences with unknown antigen specificity. *Bioinformatics*, **37**, 4865–4867.

Vander Heiden,J.A. AIRR Community. *et al.* (2018) AIRR community standardized representations for annotated immune repertoires. *Front. Immunol.*, **9**, 2206.

Weber,C.R. *et al.* (2022) Reference-based comparison of adaptive immune receptor repertoires. *bioRxiv*, 2022.01.23.476436.

Yohannes,D.A. *et al.* (2021) Clustering based approach for population level identification of condition-associated T-cell receptor β-chain CDR3 sequences. *BMC Bioinformatics.*, **22**, 159.

Zhang,H. *et al.* (2021) GIANA allows computationally-efficient TCR clustering and multi-disease repertoire classification by isometric transformation. *Nat. Commun.*, **12**, 4699.

Zobrist,A.L. (1970) A new hashing method with application for game playing. *Technical Report 88*. University of Wisconsin-Madison Department of Computer Sciences.

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btac505/6664651 by University of Oslo user on 16 August 2022