Peer Community In Genomics

A unique and customizable approach for functionally annotating prokaryotic genomes

Gavin Douglas *based on peer reviews by* ***Kwee Boon Brandon Seah*** *and* ***Max Emil Schön***

Bertrand Néron, Rémi Denise, Charles Coluzzi, Marie Touchon, Eduardo P. C. Rocha, Sophie S. Abby (2023) MacSyFinder v2: Improved modelling and search engine to identify molecular systems in genomes. bioRxiv, ver. 2, peer-reviewed and recommended by Peer Community in Genomics. https://doi.org/10.1101/2022.09.02.506364

Submitted: 09 September 2022, Recommended: 24 February 2023

Cite this recommendation as:
Douglas, G. (2023) A unique and customizable approach for functionally annotating prokaryotic genomes. *Peer Community in Genomics*, 100233. https://doi.org/10.24072/pci.genomics.100233

Published: 24 February 2023

Macromolecular System Finder (MacSyFinder) v2 (Néron et al., 2023) is a newly updated approach for performing functional annotation of prokaryotic genomes (Abby et al., 2014). This tool parses an input file of protein sequences from a single genome (either ordered by genome location or unordered) and identifies the presence of specific cellular functions (referred to as “systems”). These systems are called based on two criteria: (1) that the ”quorum” of a minimum set of core proteins involved is reached the “quorum” of a minimum set of core proteins being involved that are present, and (2) that the genes encoding these proteins are in the expected genomic organization (e.g., within the same order in an operon), when ordered data is provided. I believe the MacSyFinder approach represents an improvement over more commonly used methods exactly because it can incorporate such information on genomic organization, and also because it is more customizable.

Before properly appreciating these points, it is worth noting the norms and key challenges surrounding high-throughput functional annotation of prokaryotic genomes. Genome sequences are being added to online repositories at increasing rates, which has led to an enormous amount of bacterial genome diversity available to investigate (Altermann et al., 2022). A key aspect of understanding this diversity is the functional annotation step, which enables genes to be grouped into more biologically interpretable categories. For instance, gene calls can be mapped against existing Clusters of Orthologous Genes, which are themselves grouped into general categories such as ‘Transcription’ and ‘Lipid metabolism’ (Galperin et al., 2021).

This approach is valuable but is primarily used for global summaries of functional annotations within a genome: for example, it could be useful to know that a genome is particularly enriched for genes involved in lipid metabolism. However, knowing that a particular gene is involved in the general process of lipid metabolism is less likely to be actionable. In other words, the desired specificity of a gene's functional annotation will depend on the exact question being investigated. There is no shortage of functional ontologies in genomics that can be applied for this purpose (Douglas and Langille, 2021), and researchers are often overwhelmed by the choice of which functional ontology to use. In this context, giving researchers the ability to precisely specify the gene families and operon structures they are interested in identifying across genomes provides useful control over what precise functions they are profiling. Of course, most researchers will lack the information and/or expertise to fully take advantage of MacSyFinder's customizable features, but having this option for specialized purposes is valuable.

The other MacSyFinder feature that I find especially noteworthy is that it can incorporate genomic organization (e.g., of genes ordered in operons) when calling systems. This is a rare feature among commonly used tools for functional annotation and likely results in much higher specificity. As the authors note, this capability makes the co-occurrence of paralogs, and other divergent genes that share sequence similarity, to contribute less noise (i.e., they result in fewer false positive calls).

It is important to emphasize that these features are not new additions in MacSyFinder v2, but there are many other valuable changes. Most practically, this release is written in Python 3, rather than the obsolete Python 2.7, and was made more computationally efficient, which will enable MacSyFinder to be more widely used and more easily maintained moving forward. In addition, the search algorithm for analyzing individual proteins was fundamentally updated as well. The authors show that their improvements to the search algorithm result in an 8% and 20% increase in the number of identified calls for single and multi-locus secretion systems, respectively. Taken together, MacSyFinder v2 represents both practical and scientific improvements over the previous version, which will be of great value to the field.

References:

Abby SS, Néron B, Ménager H, Touchon M, Rocha EPC (2014) MacSyFinder: A Program to Mine Genomes for Molecular Systems with an Application to CRISPR-Cas Systems. PLOS ONE, 9, e110726. https://doi.org/10.1371/journal.pone.0110726

Altermann E, Tegetmeyer HE, Chanyi RM (2022) The evolution of bacterial genome assemblies - where do we need to go next? Microbiome Research Reports, 1, 15. https://doi.org/10.20517/mrr.2022.02

Douglas GM, Langille MGI (2021) A primer and discussion on DNA-based microbiome data and related bioinformatics analyses. Peer Community Journal, 1. https://doi.org/10.24072/pcjournal.2

Galperin MY, Wolf YI, Makarova KS, Vera Alvarez R, Landsman D, Koonin EV (2021) COG database update: focus on microbial diversity, model organisms, and widespread pathogens. Nucleic Acids Research, 49, D274–D281. https://doi.org/10.1093/nar/gkaa1018

Néron B, Denise R, Coluzzi C, Touchon M, Rocha EPC, Abby SS (2023) MacSyFinder v2: Improved modelling and search engine to identify molecular systems in genomes. bioRxiv, 2022.09.02.506364, ver. 2 peer-reviewed and recommended by Peer Community in Genomics. https://doi.org/10.1101/2022.09.02.506364

Reviews

Evaluation round #2

Reviewed by Kwee Boon Brandon Seah, 31 January 2023

I thank the authors for their detailed responses and revisions. In my opinion, they have addressed the points raised in the previous reviews. The new sections comparing MacSyFinder v1 and v2 make it clearer to understand what changes were made to the method and how this improves the predictions.

One minor suggestion: In Figure 4A and B, the number predicted per system is shown with a line graph, but a bar graph would be more appropriate because the horizontal axis is categorical.

Otherwise I have nothing further to add.

Reviewed by Max Emil Schön, 05 February 2023

I think the authors did a good job in addressing all concern that were raised by me, the other reviewer and the recommender. Specifically, I think the manuscript is now clearer and the revisions will help new users of the tool to use it more efficiently on their own datasets as well as extend it with further models. The text now also includes important sections where the two versions of the software are compared to each other and where the issues of using different inputs such as MAGs/SAGs are discussed.

I have no additional comments and would therefore support the recommendation of the article in its current form.

Evaluation round #1

DOI or URL of the preprint: `https://www.biorxiv.org/content/10.1101/2022.09.02.506364v1`

Authors' reply, 22 January 2023

Download author's reply
Download tracked changes file

Decision by Gavin Douglas, posted 12 October 2022

Revisions required

To Dr. Abby and colleagues, Two reviewers have examined your manuscript and they both agree it is interesting and is of value for the field, which I also agree with. Their overlapping comments pertain to clarifying certain points of confusion, through adding additional text and figures where needed, and also clarifying how metagenome-assembled genomes could be used as input. I also have some comments in addition to what the reviewers brought up, mainly related to adding additional clarification. Please see below. I think after addressing all of our combined comments that your manuscript will be clearer, particularly for users new to the MacSyFinder framework. Please let me know if you need clarification on any requested changes.
All the best, Gavin Douglas —-
Recommender's comments Major I think reviewer #2 had great suggestions for future development. I think they are beyond the scope of this specific manuscript though, with the exception of providing some example input and output files. It would be helpful for users to have these example files so they can try running the tool themselves and confirm they are getting the right output. Ideally, this would be for all common types of input formats. One simple way of doing this would be to upload some example input/output files to FigShare and then add some example usage commands using these files to the tool README. How long does the tool take to run on typical input files and how much memory does it use? Can it be run on multiple cores, and if so, roughly what is the relative increase in run-time (e.g., is it linear)?
I think the manuscript would be greatly improved if a brief discussion was added on the following three general areas. I think a short section could be added to the discussion/results to talk about these caveats (if they

are indeed caveats). First, how sensitive are the results are to different default values for the scores (e.g., as mentioned at L320). Does tweaking these parameters typically make much difference? Also, how were the default scores chosen? It sounds like they were selected to produce reasonable results on datasets where specific systems are known to be present, but that's not totally clear. As a potential user I would be worried that these default values would create biases when applied to genomes with different characteristics than in the training datasets. I'm guessing this kind of investigation was important when developing the first version of the tool, or for one of the other related manuscripts, but either way it would be good to quickly refer to any past validations for readers unfamiliar with the earlier work. Second, and similarly, how confident can you be in any definitions of co-localization for specific systems across different lineages? My concern is that these co-localization relationships may be well-described in certain lineages, but that operon structure may be less in others, and so these tools might not work as well. Do you think this is a concern? And if so, what should users watch out for? Last, how many systems are affected by the choice of whether components are allowed to overlap across systems are not (e.g., L330 and the examples later)? Knowing this would help give users an idea of whether it's worth exploring the multi_system/multi_model options. It's also not totally clear to me how much one might expect the overall results to change when using those options.
Minor The tool's link should be added `https://github.com/gem-pasteur/macsyfinder` to the end of the abstract "Nanomachines" is not a commonly used term in this context (to my knowledge at least), so I recommend that the authors define it or use a simpler term In the intro it is implied that "components" is synonymous with "proteins". If this is the case, then I think the authors should explicitly explain why they opted for this term over simply saying proteins. I'm thinking that perhaps the authors actually mean that components can be non-protein coding genes and perhaps noncoding elements as well, which would explain why this more general term is used. Either way, this should be explained (but based on the CRISPR-Cas section I don't think this is the case). L33 – rather than "performing function of interest" I would reword to be "performing the same function as this system" L34 – rather than "coherent" I would say "distinct, non-co-localized" perhaps to be clearer? (If that is what you mean there) L67 – I think the limitations should be in a separate paragraph, as they are the key motivation for making the new version, so you want to make readers don't miss them. L70 – Unclear what "component-specific filtering criteria" refer to here, without looking at methods. It would be nice to see a quick example. L77 – Instead of "novel" I think you mean "new" L82 – should be "takes" rather than "gets" L83 – "fasta" should be "FASTA", upon all usages L102 – change "have" to "had" L120, L476 – change "..." to ", etc." L158: A quick description of what GA scores are based on (or an example) would be useful E-values are well known, but I haven't seen the term "i-evalue" before. Maybe I missed where it was defined, but if not this should be defined and distinguished from normal E-values. L328 – should be "are" rather than "were" for both instances (i.e., use present tense when describing what the tool does in general, and not just what it did on a single occasion) L335-336: It's not clear to me what the edges are based on in this case (i.e., do they represent the binary relationship of whether the systems can be compatible or not? Or can they be weighted?). I think it would be useful to add a sentence clarifying that for potential users who aren't familiar with the clique search approach. L393 – Capitalize "archaea" and "bacteria" (here and elsewhere) to be consistent with earlier usage L468 – rather than "we present its application" I would say "we apply it"
L499 – Make "Multi_model" lowercase (since I'm guessing the module is case sensitive?) L524: "that the same cluster is" should be "for the same cluster to be"

Reviewed by Max Emil Schön, 26 September 2022

First of all congrats to the authors for a very nice tool and an interesting corresponding paper! I do not have many comments, as I think the usefulness and impact of the tool is apparent and the paper presents its main points clearly and concisely. Importantly, all described updates and improvements are well justified and will likely help in spreading the application of MacSyFinder and improve its predictions. The addition of a dedicated structure and automatic handling of models is also very handy and should significantly lower the threshold for new users, as will the intuitive installation process via pip, conda or docker. The available online documentation is very extensive, detailed and well structured. The application of GA scores (which I did not know about before) is very elegant and should reduce false positives. The ability to re-analyze runs using macsyprofile could be very useful for in-depth searches for e.g. novel variations of molecular systems.

Below are my comments on the paper, please consider them suggestions. I would be very interested to see your comments on these issues, but I think the paper could be considered complete even without theses additions.

general comments

While the authors describe the improvements that were made between v1 and v2, I think it would be great to see some visualizations/data substantiating this. I could imagine e.g. an additional figure showing the discovery of a system that was not detected in v1 but now is found by v2. Information on the number of false positive/false negative detections, on the gene or system level, for both versions on one or several example genomes could be interesting as well. Showing this in a figure would help the reader appreciate at a glance why the development of such a major update was indeed necessary.

Have you assessed the impact of fragmented assemblies (e.g. incomplete MAGs) on the performance of MacSyFinder v2? I think it could be worthwhile to many readers how this affects the prediction, since many people will likely apply it on genomes that are, to a certrain degree, incomplete or fragmented. I am for example thinking of systems that are separated by contig boundaries and therefore potentially not in the correct order as they are on the genome.

specific comments

abstract: I think it's worth mentioning here that the target organisms are only bacteria and archaea, but not eukaryotic microbes.

L 15-16: I find this sentence a bit hard to parse, consider clarifying it.

L 109: Is there some sort of XML schema for validating the model definition in the new macsy-model packages? This could be useful for future modellers.

L 172-177: A somehwat difficult/unclear paragraph that could perhaps be clarified.

L 219: There's something wrong with this sentence I think.

Table 1: Consider changing 'Nb' to 'No.'

L 385-389: This sentence could be rephrased for more clarity.

Macsy-models github organization: link to `https://github.com/macsy-models/.github/blob/CONTRIBUTING.md` is broken

Reviewed by Kwee Boon Brandon Seah, 12 October 2022

This manuscript (`https://doi.org/10.1101/2022.09.02.506364`) describes a new version (v2) of the tool MacSyFinder, which searches for gene clusters in microbial genomes (e.g. coding for macromolecular complexes) by first using HMMs to identify individual protein components, and then user-defined models of essential and accessory components to annotate the clusters. In this version, the code was updated to Python 3, the modeling and search engine were improved, and new tools were added to make it easier to distribute and install models.

The previous version of MacSyFinder (`https://doi.org/10.1371/journal.pone.0110726`) and tools in which it has been integrated, such as CrisprCasFinder, appear to be popular and widely adopted. The updates should improve the user-friendliness of the software.

Specifically, the software is straightforward to install via different distribution channels (Conda, pip, Docker container). The macsydata tool and the use of Git repositories to distribute models is convenient for both users and modellers, and is a good idea for getting more community participation and to make the tool more extensible. Extensive documentation for users, modellers, and developers is available online. Several of the most often used models such as those for CRISPR-Cas systems have been ported to MacSyFinder v2.Major commentsComparison to other tools

Could the authors briefly discuss the intended use cases for MacSyFinder and how they differ from other tools for pathway and gene cluster annotation, such as KEGG Mapper, Pathway Tools, and Antismash (secondary metabolite gene clusters)? For example, the logical expressions used to define KEGG Modules has similarities to the model definitions in MacSyFinder, but don't incorporate information about collocation, as far as I know. This could help give some context for readers and users.Input data formats

It appears to be only possible to analyse either complete closed genomes where the gene order is known (in ordered replicon mode), or treat each gene independently (unordered mode), which in the v1 paper was suggested for the analysis of metagenomes. However it is now quite common to work with draft genomes and metagenome assembled genomes (MAGs) where the gene order is only partly known. Is it possible to exploit this partial information? In the user guide, a third input option, ”gembase” is offered as a solution for analyzing multiple genomes at once, but this is not mentioned in the paper. Would it be appropriate to use this input mode for draft genomes?Improvements to search engine

My expertise is not in algorithm design so I can't comment too much on the method itself, but could the authors give an example of a suboptimal solution found by v1 compared to the improved results from v2 (see lines 431-465)? It would be helpful to have a concrete example of what these look like in v1 so that we can see how v2 produces better results.

I also found it difficult to understand the description of the heuristic used by v2 to simplify out-of-cluster components (lines 454-456); perhaps a diagram might be helpful here. The results shown in Figure 4 appear to be the output from v2 only.Predicting complete vs. incomplete systems

An incomplete system could be missing one or more of its components, whether mandatory or accessory. In the example shown in Figure 5C (dCONJ typeF), one of the three original mandatory genes (traI) is still mandatory even though the model is incomplete. However, isn't it possible to have an incomplete copy of the T4SS where only traI is missing? Could one define a model for incomplete systems simply by changing all genes that are mandatory to ”accessory”?Suggestions for future development

Finally I have a few suggestions for future versions of the software that the authors could consider.

- Bundle some example data for users to test the software after installation, and also to help them learn how to use the various options.
- In addition to the model validator macsydata check , how about something like macsydata init to set up the directory structure and template files to get started with a model?
- Allow annotated nucleotide sequences as input too, e.g. feature tables + nucleotide Fasta + genetic code, or EMBL/Genbank files. This would simplify the specification of feature coordinates, e.g. when we are working with draft genomes where a complete replicon is not available, only individual contigs.

Minor comments

- Figure 1. Panels 3.1 and 3.2 ”exam” should probably be ”examination” or ”examine”
- Line 219. I think the authors mean ”ported to Python 3” instead of ”carried under Python 3”.
- Line 295. Suggest to change ”xm” and ”xa” to use subscripts for m and a, to avoid giving the impression that these are products x times m or x times a.
- Lines 338-339. Could the authors clarify what ”this” in ”this is the definition of a clique” refers to? Is it referring to the set of connected systems or the connections between them (or are these equivalent?).
- Line 446. Which panel in Figure 4 is being referred to?

- Line 583. The word ”degenerate” here appears to be used here in the sense of ”decayed”, ”degraded”, or ”incomplete”. However it can easily be confused with the technical meaning of ”having multiple elements that correspond to a single element”, especially in the context of defining a model. I suggest sticking to ”complete” vs. ”incomplete” or similar terminology used elsewhere in the manuscript.
- The software is archived at the Software Heritage Archive, but this isn’t mentioned in the manuscript. I suggest also citing the archive DOI.