



**HAL**  
open science

## MacSyFinder v2: Improved modelling and search engine to identify molecular systems in genomes

Bertrand Néron, Rémi Denise, Charles Coluzzi, Marie Touchon, Eduardo P.C.  
Rocha, Sophie S Abby

### ► To cite this version:

Bertrand Néron, Rémi Denise, Charles Coluzzi, Marie Touchon, Eduardo P.C. Rocha, et al.. Mac-SyFinder v2: Improved modelling and search engine to identify molecular systems in genomes. 2022. pasteur-04060331v1

**HAL Id: pasteur-04060331**

**<https://hal.science/pasteur-04060331v1>**

Preprint submitted on 21 Oct 2022 (v1), last revised 6 Apr 2023 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

# MacSyFinder v2: Improved modelling and search engine to identify molecular systems in genomes

Bertrand Néron<sup>1</sup>, Rémi Denise<sup>2,3</sup>, Charles Coluzzi<sup>2</sup>, Marie Touchon<sup>2</sup>, Eduardo P.C. Rocha<sup>2,\*</sup>, Sophie S. Abby<sup>4,\*</sup>

<sup>1</sup>Institut Pasteur, Université Paris Cité, Bioinformatics and Biostatistics HUB, Paris, France

<sup>2</sup>Institut Pasteur, Université Paris Cité, CNRS UMR3525, Microbial Evolutionary Genomics, Paris, France

<sup>3</sup>APC Microbiome Ireland & School of Microbiology, University College Cork, Cork, Ireland

<sup>4</sup>Univ. Grenoble Alpes, CNRS, UMR 5525, VetAgro Sup, Grenoble INP, TIMC, 38000 Grenoble, France

\*corresponding authors

## **Email addresses:**

Bertrand Néron: [bneron@pasteur.fr](mailto:bneron@pasteur.fr)

Eduardo Rocha: [erocha@pasteur.fr](mailto:erocha@pasteur.fr)

Sophie Abby: [sophie.abby@univ-grenoble-alpes.fr](mailto:sophie.abby@univ-grenoble-alpes.fr)

## **ORCID numbers:**

Bertrand Néron: [0000-0002-0220-0482](https://orcid.org/0000-0002-0220-0482)

Rémi Denise: [0000-0003-2277-689X](https://orcid.org/0000-0003-2277-689X)

Charles Coluzzi: [0000-0003-2238-0836](https://orcid.org/0000-0003-2238-0836)

Marie Touchon: [0000-0001-7389-447X](https://orcid.org/0000-0001-7389-447X)

Eduardo PC Rocha: [0000-0001-7704-822X](https://orcid.org/0000-0001-7704-822X)

Sophie S Abby: [0000-0002-5231-3346](https://orcid.org/0000-0002-5231-3346)

1 **ABSTRACT**

2

3 Complex cellular functions are usually encoded by a set of genes in one or a few  
4 organized genetic loci in microbial genomes. MacSyFinder uses these properties to  
5 model and then annotate cellular functions in microbial genomes. This is done by  
6 integrating the identification of each individual gene at the level of the molecular  
7 system. We hereby present a major release of MacSyFinder (Macromolecular System  
8 Finder), MacSyFinder version 2 (v2). This new version is coded in Python 3 ( $\geq 3.7$ ).  
9 The code was improved and rationalized to facilitate future maintainability. Several new  
10 features were added to allow more flexible modelling of the systems. We introduce a  
11 more intuitive and comprehensive search engine to identify all the best candidate  
12 systems and sub-optimal ones that respect the models' constraints. We also introduce  
13 the novel *macsydata* companion tool that enables the easy installation and broad  
14 distribution of the models developed for MacSyFinder (macsy-models) from GitHub  
15 repositories. Finally, we have updated, improved, and made available MacSyFinder  
16 popular models for this novel version: TXSScan to identify protein secretion systems,  
17 TFFscan to identify type IV filaments, CONJscan to identify conjugative systems, and  
18 CasFinder to identify CRISPR associated proteins.

## 19 INTRODUCTION

20

21 Microbial nanomachines and pathways (hereafter called “systems”) can be very  
22 complex and involve many proteins (hereafter called “components”). In the genomes  
23 of Bacteria and Archaea, the components of these systems are often encoded in a  
24 highly organized way, involving one or a few operons with functionally related genes.  
25 For example, loci encoding the peptides of a protein complex or the enzymes of a  
26 metabolic pathway have specific genetic organizations that tend to be remarkably  
27 conserved (Dandekar et al., 1998; Teichmann and Babu, 2002). Neighbouring operons  
28 in genomes are also often functionally related (Huynen et al., 2000). This means that  
29 gene co-localization can be used to infer gene functions and improve homology  
30 inference, e.g., when sequence similarity is low. Co-localization also facilitates the  
31 distinction between functionally diverging homologs (Abby and Rocha, 2012). The  
32 hypothesis is that the member of the protein family that co-localizes with the rest of the  
33 system's genes is the one performing the function of interest. Conversely, many  
34 cellular processes require an ensemble of coherent components. In such cases, a  
35 function can only be identified when the repertoire of genes is analyzed at the system-  
36 level. For example, a minimum set of components is necessary for the functioning of a  
37 protein secretion system.

38

39 In 2014, we published the “Macromolecular System Finder” (MacSyFinder v1) program  
40 for the functional annotation of cellular machineries and metabolic pathways in  
41 microbial genomes (Abby et al., 2014). It makes a system-level annotation that takes  
42 advantage of the typical functional organization of microbial genomes (co-localization)  
43 and the requirement of a core set of protein components to perform the function  
44 (quorum). MacSyFinder consists of a generic modelling framework and a search  
45 engine to screen genomes for candidate systems. The modelling framework enables  
46 the user to define models for the systems of interest, including the components'  
47 identity, category, and genetic organization. MacSyFinder v1 has three categories of  
48 components: mandatory, accessory, and forbidden. Parameters of gene co-  
49 localization describe the genomic architecture of the system at the level of each  
50 component or of the entire system. Each component corresponds to one HMM (hidden  
51 Markov model) profile to enable sequence similarity search with HMMER (Eddy, 2011)  
52 and different components (and thus profiles) can be defined as exchangeable if they  
53 have the same role in the system. The search engine screens a database of genomes  
54 for potential systems using HMM profile searches and the clustering of co-localized  
55 hits along the genome that match the systems' model.

56

57 MacSyFinder has been used with success to annotate a variety of microbial  
58 machineries and pathways, including protein secretion systems (Abby et al., 2016),  
59 CRISPR-Cas systems (Abby et al., 2014; Couvin et al., 2018) and other prokaryotic  
60 defence systems (Tesson et al., 2022), capsular loci (Rendueles et al., 2017), DNA  
61 conjugation systems (Cury et al., 2020), the butyrate production pathway (Sharp and  
62 Foster, 2022), methanogenic and methylotrophic metabolisms (Adam et al., 2019;  
63 Chibani et al., 2022) , cell division machineries (Pende et al., 2021) and outer  
64 membrane protein clusters (Taib et al., 2020). It has enabled wide-scale genomic  
65 analyses of biologically relevant systems and was integrated into the popular  
66 MicroScope genome annotation pipeline and in the reference CRISPRCasFinder  
67 program (Couvin et al., 2018; Vallenet et al., 2020). Yet it has several limitations. In

68 terms of software engineering, it is coded in the now obsolete Python v2.7, lacks tools  
69 to improve its future development and maintenance, and some parts of the program  
70 are not efficient. In terms of modelling, it cannot use component-specific filtering criteria  
71 to annotate the genes. Furthermore, it lacks a way to annotate genes of interest that  
72 are neutral concerning the systems' assessment. More importantly, the greedy search  
73 engine is not optimal and has complex, sometimes counter-intuitive behaviours.

74  
75 We hereby present a major release of MacSyFinder, MacSyFinder version 2 (v2)  
76 coded in Python 3 ( $\geq 3.7$ ). In addition, we have updated and improved the most  
77 popular MacSyFinder models to the novel version to make them readily usable.

## 78 MATERIALS AND METHODS

79

### 80 • Input & Output files

81

82 MacSyFinder v2, like v1, gets as input files the models of the systems to search and a  
83 multi-protein fasta file (Figure 1). When the proteins are ordered in the file as in the  
84 replicon, one can use the most powerful search mode – “ordered\_replicon” – to study  
85 the genetic organization. Otherwise, the search mode “unordered” must be used. The  
86 significant modifications in v2 concern the organization of the input systems’ models  
87 (“macy-model” packages, see below) and the output files. The latter were adapted to  
88 reflect the new MacSyFinder search engine results. In addition, various easy-to-parse  
89 text tabulated files are now proposed, including the raw and filtered results of the  
90 components’ similarity search with HMMER, the component-wise description of the  
91 possible systems, the systems constituting the best solutions, and the component-wise  
92 description of rejected candidates. For more details, one can consult MacSyFinder’s  
93 comprehensive documentation, including the User Guide, the Modeller Guide, and the  
94 Developer Guide, created with Sphinx and available at:  
95 <https://macyfinder.readthedocs.io/>.

96

### 97 • Formalizing macy-model packages

98

99 MacSyFinder v1 required two directories containing one or several systems’ models  
100 (“definitions” folder) and the corresponding HMM profiles (“profiles” folder). These files  
101 were passed to the command line as mandatory parameters and were distributed as  
102 standalone archives. Unfortunately, these have the inconvenience of being poorly  
103 versionable or traceable. To improve the reproducibility of analyses with MacSyFinder  
104 v2, we increased the traceability of the models and facilitated their retrieval and  
105 installation by formalizing a package structure that we call “macy-model” package  
106 (see Fig. S1).

107 A macy-model package must have two directories: “definitions” and “profiles”. The  
108 “definitions” directory contains all model definitions written in the MacSyFinder-specific  
109 XML grammar (one file per model definition). This directory can include several sub-  
110 directories and levels (Fig. S1). The “profiles” directory contains all HMM protein  
111 profiles appearing in the definitions. In addition, a new file, “*metadata.yml*”, was  
112 introduced to store necessary metadata such as the package name, version,  
113 description, citation, distribution license, and the contacts of its author(s)/maintainer(s).  
114 Some facultative but recommended files can be added: LICENSE/copying,  
115 Contributing, README.md, and model\_conf.xml. The README file should explain  
116 how to use the models and can be displayed using the command *macydata help* (see  
117 below). The file *model\_conf.xml* allows the modeller to set package-specific  
118 configurations such as score configuration options (see paragraph on scoring) or  
119 criteria to filter the hits when searching the components (profile coverage threshold,  
120 usage of GA scores with HMMER...). The user can easily supersede these  
121 recommended values using the command line and configuration files.

122

### 123 • Grammar update for the modelling framework

124

125 The models of MacSyFinder are written using a dedicated XML grammar with a

126 hierarchy that fits the hierarchical nature of the biological systems to model: systems'  
127 models are made of gene components (Fig. 3, Supplementary Table 1). The two main  
128 objects in the hierarchy of a system's model are thus the "model" (replacing the  
129 "system" keyword in v1) at the top level and the "gene" at the lower level. In addition,  
130 a feature "vers" was added at the model level to indicate the version of the grammar:  
131 "vers=2.0" matches MacSyFinder v2.

132 We simplified the XML grammar to ensure better readability and easier maintenance  
133 of the models (Fig. 3, Supplementary Table 1). Relative to the first version, some  
134 keywords were removed or merged into novel ones. This is the case of the keywords  
135 "homologs" and "analogs" that were replaced by the new keyword "exchangeables" to  
136 indicate that some components can be "exchanged" by others (*i.e.*, fill the same role  
137 in systems). The gene attribute "exchangeable" was thus removed as not needed  
138 anymore. The "system\_ref" keyword was also removed.

139 When designing a system's model or investigating the distribution of genes within  
140 genomic occurrences of a system, one may want to annotate genes that are not  
141 important to the identification/discrimination of the system. We introduced a new type  
142 of component called "neutral". It adds to the pre-existing mandatory, accessory, and  
143 forbidden components, except that this new type is not used to score the systems or  
144 assess their quorum (minimal number of components required in a system). Neutral  
145 components are identified using HMM protein profiles and placed into clusters like the  
146 other components. Hence, even if they do not contribute to the scoring of systems,  
147 they can "connect" or "extend" clusters of mandatory and accessory components.  
148 Details, examples, and a tutorial concerning the XML grammar v2 are available in  
149 MacSyFinder's documentation, which now includes a novel section called the  
150 "Modeller Guide" to explain how to build novel models of systems. We also provide  
151 some of the most popular models translated with improvements for MacSyFinder v2  
152 (Table 1). They are readily usable and installable through the *macsydata* program (see  
153 Results section).

154  
155 • **Enabling component-wise filtering by setting up GA scores**

156  
157 The search for components can now use the "GA" (Gathering) scores of the HMM  
158 profiles. This allows using component-wise criteria for hit filtering instead of having the  
159 same criteria for all components (as in v1 of MacSyFinder). If a GA score is present in  
160 the HMM profile file, the system calls HMMER using the option "--cut\_ga", which  
161 supersedes the *i*-value and profile coverage default values otherwise used in the  
162 absence of GA scores. It is possible to deactivate the GA scores using the new option  
163 "--no-cut-ga" (False by default). The rules for the filtering of the components can be  
164 specified (by decreasing order of priority): in the command line, in the model  
165 configuration file of the system ("model\_conf.xml"), or using the HMM protein profile  
166 GA scores (or the "i-value" and "profile coverage" if GA scores not provided).

167  
168 Many of the HMM protein profiles used in MacSyFinder models are from PFAM or  
169 TIGRFam (Sonnhammer et al., 1997; Haft et al., 2003) and already include GA  
170 thresholds. To enable users to efficiently use the profiles previously developed for  
171 MacSyFinder v1 models, we computed the GA scores for CasFinder, TXSScan,  
172 CONJScan, and TFFscan profiles (see Table 1). To this end, we annotated with the  
173 corresponding models the completely assembled genomes of 21105 bacterial and

174 archaeal strains retrieved from the non-redundant NCBI RefSeq database (as of  
175 March, 2021). We analysed the distribution of the scores for the hits for the different  
176 genes found in the detected systems, and attributed as GA score the minimal score  
177 observed to the corresponding profile.

178 In total, 1000 HMM profiles from the four packages are available with GA scores. They  
179 thus now offer the possibility for component-wise filtering with HMMER, ensuring  
180 optimal usage of the 104 macy-s-models hereby updated for this new version of  
181 MacSyFinder.

182

183 • **Sharing and handling macy-s-models with the *macy-sdata* command and**  
184 **the “MacSy Models” organization**

185

186 The novel tool “*macy-sdata*” was created to make macy-s-model packages easily  
187 traceable, versionable, shareable, and automatically installable. It was designed to be  
188 as light as possible for the modellers and was inspired by the packaging workflows  
189 found in some Linux distributions such as Gentoo (<https://www.gentoo.org>). In addition,  
190 the *macy-sdata* API was inspired by *pip*, which is familiar to most Python users.  
191 *macy-sdata* implements common sub-commands such as *search*, *install*, *upgrade*,  
192 *uninstall*, and *help* (see Table 2). We also implemented some specific useful sub-  
193 commands for MacSyFinder, such as *cite* to display how to cite the macy-s-model  
194 package and *definition* to show a set of models’ definitions in XML format.

195 The “MacSy Models” Github organization was designed to serve as an umbrella  
196 organization to host any macy-s-model package. It allows the modeller to distribute their  
197 packages to all MacSyFinder v2 users efficiently. First, the modeller creates a *git*  
198 repository. Then, the quality of the package (package structure, model definitions  
199 syntax, the coherence between definitions and profiles) can be checked using the  
200 *macy-sdata check* command on the directory containing the entire file architecture of a  
201 macy-s-model package (Fig. S1). Finally, when everything is up to standards, the  
202 modeller has just to tag the repository and push it under the Github organization  
203 “MacSy Models”. This action allows the model package to be visible from the  
204 *macy-sdata search* tool and thus findable and accessible for remote installation to all  
205 MacSyFinder users. *macy-sdata* uses the Github Rest API to search and download the  
206 packages. Of note, *macy-sdata* can also install macy-s-model packages from a tarball  
207 archive, given it respects the above-described file architecture.

208

209 • **The macy-sprofile companion tool**

210

211 The novel tool “*macy-sprofile*” of the MacSyFinder suite allows filtering and extracting  
212 HMMER hits with settings different from those used during the run. This allows  
213 retrieving relevant hits not initially included in predicted systems, e.g., to understand  
214 why they were “missed”. This could be particularly useful to assist the design of the  
215 profiles and the systems’ models or to search for atypical versions of the systems (see  
216 details in the online documentation).

217

218 • **Code implementation, dependencies, and availability**

219 The code was carried under Python 3 ( $\geq 3.7$ ). Many unit and functional tests were  
220 implemented to reach a coverage of the code of 97%. The program requires the  
221 HMMER suite ( $\geq 3.1b2$ ) for the search of components and several well-established



222 and stable Python libraries to facilitate models' packaging (*pyyaml*, *packaging*), deal  
223 with output files (*colorlog*, [pandas](#)), and search for the best solution (*NetworkX*, see  
224 below).

225 The code and macy-model packages are available on Github under the GPL v3  
226 license: <https://github.com/gem-pasteur/macyfinder> and [https://github.com/macy-](https://github.com/macy-models)  
227 [models](#). In addition, a pypi package, a conda package and a Docker container were  
228 created to enable the easy deployment of the MacSyFinder suite.

229

## 230 **THE MACSYFINDER V2 SEARCH ENGINE**

231

232 An overview of the new search engine is provided in Figure 1. The first steps of  
233 MacSyFinder v2 search engine remain mostly unchanged relative to v1. First, it uses  
234 HMMER to search for occurrences of the non-redundant components listed in the  
235 models in the input genome(s). The best hits are assigned to the corresponding  
236 components, and are filtered by profile coverage (>50% by default) and i-evalue  
237 (<0.001 by default) when no GA score is available for the profiles.

238

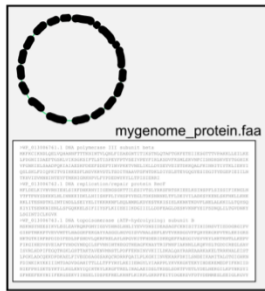
- 239 • **System-wise creation of candidate systems**

240

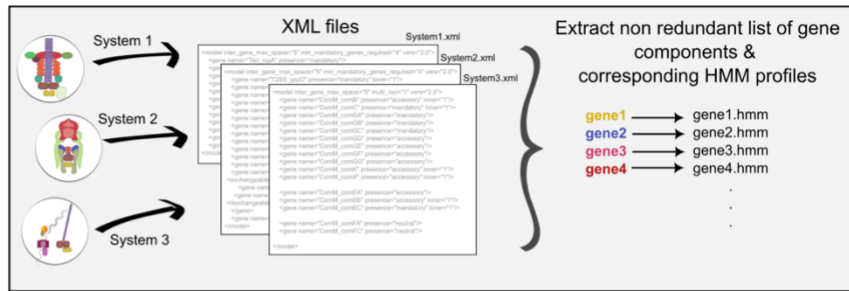
241 In version 2, the systems are searched one by one: the identified components have  
242 their hits filtered by type of system, and clusters of components are built from  
243 components respecting the co-localization criteria ("inter-gene-max-space" parameter)  
244 (Fig. 1). Candidate systems are built using the clusters of components and the  
245 components authorized to be outside of clusters ("loner" components). For "single-  
246 locus" systems, combinations of individual clusters with loner components not yet  
247 represented in the cluster are examined as candidate systems. For systems allowed  
248 to be encoded by multiple loci, all possible combinations of identified clusters and loner  
249 components (not found in clusters) are assessed as candidate systems. The eligible  
250 systems are the candidate systems that respect the systems' model in terms of the  
251 minimal quorum criteria for all components and the mandatory ones. The other  
252 systems are rejected for now and kept aside. In the case where "multi-system" genes  
253 are part of the systems' model, the list of the corresponding components will be  
254 collected from the eligible systems and combinatorically added to the set of rejected  
255 candidates to be assessed for the formation of new eligible systems.

## 1. Input files

genome to analyse

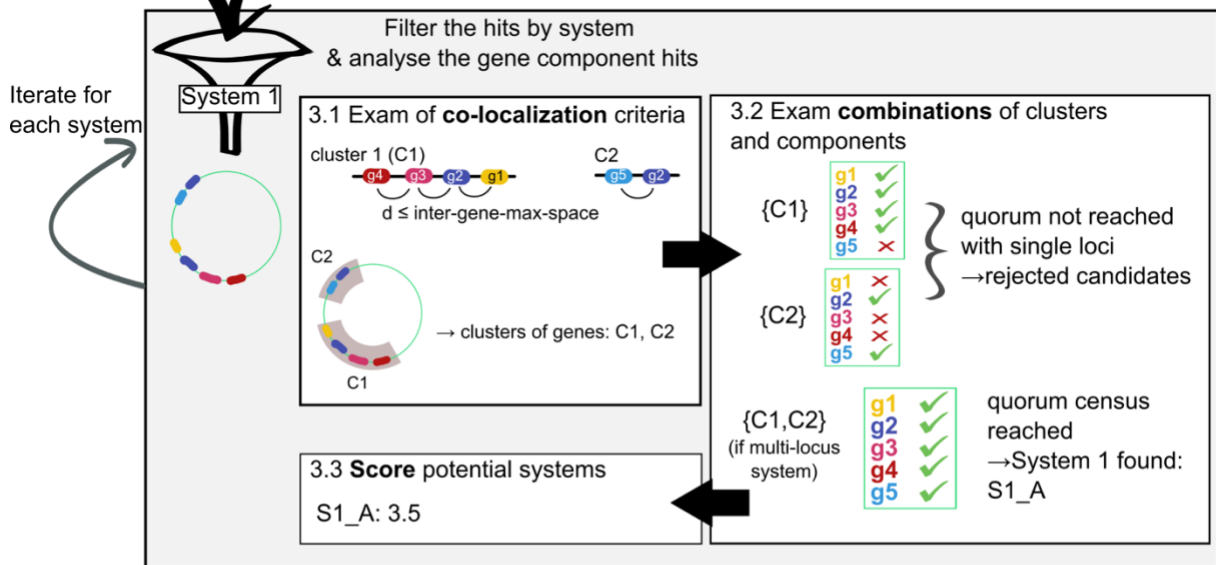


systems to detect (macy-model package)

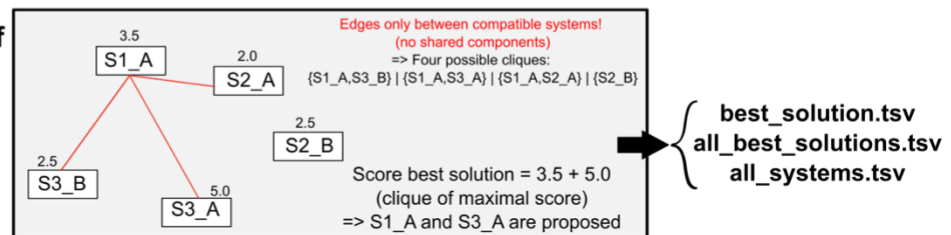


## 2. Search genome for the best hits for all profiles (with HMMER)

## 3. Search system by system



## 4. Build a graph of systems and find the best solution



256  
257

258

259

260

261

262

263

264

265

266

### Figure 1. Overview of the major steps of MacSyFinder v2.

(1) The user gives as input the genome(s) to analyse under the form of a multi-protein fasta file (order respecting that of the genes on genome if possible) and a macsy-model package with the systems to detect. Then the search engine establishes the non-redundant list of corresponding components. (2) The components are then searched with HMMER (hmmsearch using GA scores when available). The proteins with the best hits are filtered by i-evalue and profile coverage (if no GA score was available). (3) A system-by-system search is then performed. The hits corresponding to a first system are selected (“g1” as hit for gene 1), and clusters of components are formed by

267 gathering the hits respecting the maximal inter-gene-max-space (3.1). Components  
268 allowed to be “out-of-clusters” are also collected (loners and multi-systems). Then the  
269 possible combinations of clusters and “out-of-clusters” components are computed, and  
270 the program tests if they respect the quorum for the system (3.2). Finally, all candidate  
271 systems are scored (3.3). Step (3) is re-iterated for each system to be detected. Once  
272 all systems are examined, the best solution is searched using a graph-based  
273 approach. Step (4) A graph connecting all compatible candidate systems (i.e., systems  
274 with no shared components) is built, each node having the score of the corresponding  
275 system. The best solution is defined as the set of compatible systems obtaining the  
276 highest cumulative score. This corresponds to the clique of maximal score. Diverse  
277 output files are provided to the user, including one with the composition of (one of) the  
278 best solution, a file with all equivalent best solutions (if several reach the highest score),  
279 and one with all eligible candidate systems whether they are part of the best solution  
280 or not. Drawings of systems at Step (1) are derived from (Denise et al., 2019).

281

282 • **Introducing a scoring scheme for candidate systems**

283

284 Candidate systems that respect the quorum and co-localization conditions imposed by  
285 a system’s model are designated as eligible systems and are assigned a score (Fig.  
286 2A). The core of the system score is the sum of three terms: the sum of the scores of  
287 the  $n$  Clusters, the sum of the scores of the  $o$  out-of-cluster components (loner or multi-  
288 system, see below), plus a penalty  $P_{System}$  for the redundancy within the system.

289

290

$$S_{System} = \sum_{i=0}^n S_{Cluster\ i} + \sum_{i=0}^o S_{out-of-clust\ i} + P_{System}$$

291

292 Multiple occurrences of a component within the same cluster are counted as a single  
293 occurrence of the component. The score of each cluster is a function of the number of  
294 mandatory ( $m$ ) and accessory components ( $a$ ), and of the number of exchangeable  
295 mandatory ( $xm$ ) and exchangeable accessory ( $xa$ ) components it contains. These  
296 values are weighted to give more importance to mandatory components:  $w_{mandatory} = 1$ ,  
297  $w_{accessory} = 0.5$ , and  $w_{neutral} = 0$  (Fig. 2A). Moreover, to give more value to the originally  
298 listed gene than to the listed “exchangeables”, a factor  $f_{exchang} = 0.8$  is applied to the  
299 scores when an exchangeable gene fulfils the function. The score  $S_{Cluster}$  is then given  
300 by:

301

302

$$S_{Cluster} = m \times w_{mandatory} + f_{exchang} \cdot xm \times w_{mandatory} + a \times w_{accessory} \\ 303 \quad + f_{exchang} \cdot xa \times w_{accessory}$$

304

305 The score of the genes found outside of a system’s cluster is computed like the score  
306 of the components found in clusters “ $s_c$ ” (see above), except that a factor  $f_{out-of-clust} =$   
307  $0.7$  is applied. Here,  $s_c$  can represent any of the component-specific parts of the  $S_{Cluster}$   
308 sum presented above, depending on the mandatory, accessory, and/or exchangeable  
309 status of the out-of-cluster component:

310

311

$$S_{out-of-clust} = f_{out-of-clust} \cdot S_c$$

312

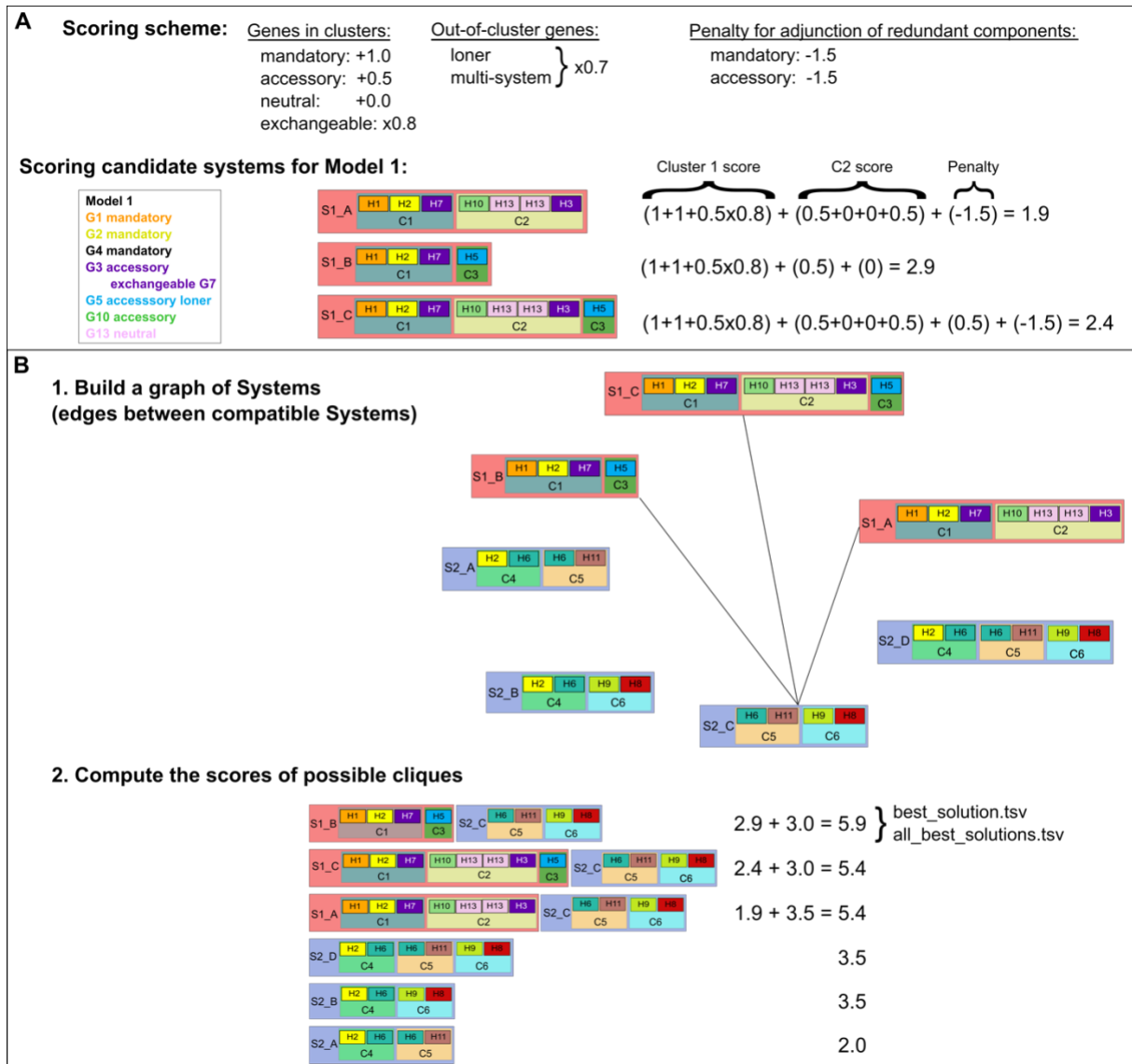
313 A component is deemed redundant only if found in more than one cluster. The penalty  
314 part of a score thus penalizes candidate systems with  $r$  redundant mandatory or  
315 accessory components, where  $r$  thus corresponds to the number of clusters with the  
316 component minus one. We define  $P_{System}$  ( $p_{redundant} = -1.5$  by default):  
317

$$P_{System} = \sum_{i=0}^r p_{redundant}$$

319  
320 The default values of the different score parts are indicative and allow MacSyFinder to  
321 behave as expected in the cases we have tested. The users can fully parameterize the  
322 weights, factors, and penalties. The modeller of a system can also ship, with a macsy-  
323 model package, its recommended values for the weights of the scoring system using  
324 the optional “*model\_conf.xml*” file.  
325

- 326 • **Combinatorial exploration of solutions**

327  
328 Once all models were searched and their occurrences were assigned scores (see  
329 above), a combinatorial examination of the possible sets of compatible systems is  
330 performed (Fig. 1 and 2B). Two systems are deemed compatible if they are made of  
331 distinct gene components. Thus, unless specified using the “*multi\_system*” or  
332 “*multi\_model*” features, a component cannot be involved in several systems. A  
333 MacSyFinder search solution is defined as a set of compatible systems. The search  
334 for the best solution corresponds to the well-known weighted maximum clique search  
335 problem (Brandes and Erlebach, 2005). The program builds a graph connecting all  
336 pairs of compatible systems, each node representing a system whose weight  
337 corresponds to its score. The goal is to identify a set of systems that are all compatible  
338 with each other, meaning that they are all inter-connected in a sub-graph. This is the  
339 definition of a “clique”. The best solution is the clique harboring the highest cumulated  
340 systems’ score, the score of a solution being the sum of the systems’ scores  
341 composing it. The “*find\_cliques*” function proposed in the NetworkX Python library is  
342 used to find the set of maximal cliques that correspond to the best possible solutions  
343 in terms of cumulated nodes’ weights of the cliques (Hagberg et al., 2008). This may  
344 result in several solutions with the maximal score, in which case they are all provided  
345 to the user (file “*all\_best\_solutions.tsv*”). The best solution, or one among the best, is  
346 given in the dedicated output file “*best\_solution.tsv*”.



347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357

**Figure 2. Scoring scheme and combinatorial search of MacSyFinder v2 search engine.** **A.** The scoring scheme is summarized, and then illustrated by an example for a hypothetical Model1. “H1” stands for a hit for gene 1 “G1” in the genome. **B.** Step (1). The graph of candidate systems is drawn by connecting all compatible systems, i.e. those with non-overlapping hits (unless authorized by the multi-model or multi-system feature). Step (2). The clique of maximal cumulated score (best solution) is searched, with the score being defined as the sum of the systems’ scores that are part of the clique. The results are stored in the files “best\_solution.tsv” and “all\_best\_solutions.tsv”, and all candidate systems are stored in “all\_systems.tsv”.

358 **RESULTS & DISCUSSION**

359

360 **I/ Grammar changes and macsy-model file architecture enable better, simpler,**  
 361 **and more intuitive systems' modelling and sharing**

362 Version 1 of MacSyFinder lacked a dedicated file architecture to share MacSyFinder's  
 363 systems' models. We now define a structured file architecture for the novel "macsy-  
 364 model packages" (see Materials and Methods and Fig. S1). In particular, there now  
 365 may be several levels of sub-directories for the "definitions" folder. This enables  
 366 running *macsyfinder* with only a pre-defined subset of models and establishing a  
 367 hierarchy of models in a biologically relevant manner. The introduction of this file  
 368 architecture thus satisfies two main objectives: it allows the file architecture of the  
 369 macsy-model packages to reflect the biological specificities of the systems while  
 370 enabling automated handling of the macsy-model packages for easier distribution and  
 371 installation via the *macsydata* tool. Several popular MacSyFinder models from v1 were  
 372 carried under MacSyFinder v2 grammar and file architecture. They are now available  
 373 at the "MacSy Models" Github organization for automated installation with  
 374 MacSyFinder v2 using the *macsydata* tool (discussed in detail below, see also  
 375 Materials and Methods, Table 1 and Table 2). The creation of the "MacSy Models"  
 376 organization enables the macsy-model packages to be versioned for better  
 377 reproducibility. This organization also constitutes the first step towards unifying a  
 378 MacSyFinder modeller community.

379

380 **Table 1. Overview of MacSyFinder v2 macsy-model packages available at the**  
 381 **"MacSy models" organization <https://github.com/macsy-models>**

Model repository	Version tag	Original reference	Systems detected	Nb models	Nb profiles	Remark
CasFinder	3.1.0	(Abby et al., 2014; Couvin et al., 2018)	CRISPR-Cas systems (Cas clusters detection, annotation, and classification)	44	535	This new version: - provides the possibility to detect more subtypes than the previous ones - greatly improves the detection of tandem systems - improves the detection of degenerated and atypical systems - can now be ran at once for the three levels of classification - GA scores added to HMM profiles
CONJscan	2.0.1	(Cury et al., 2017)	Conjugative, mobilizable and decayed conjugative systems	34	124	This new version: - allows the detection of decayed conjugative systems - provides models adapted to the detection in chromosomes or plasmids - contains tailored thresholds for each type of MPF - GA scores added to HMM profiles
TFFscan	1.0.0	(Denise et al.,	Systems members	7	169	As in the original paper, but

		2019)	of the type IV filament super-family			with GA scores added to HMM profiles
TXSScan	1.0.0	(Abby et al., 2016)	Protein secretion systems and related appendages	22	205	As in the original paper, but with GA scores added to HMM profiles
TXSScan	1.1.0	(Abby et al., 2016; Denise et al., 2019)	Protein secretion systems and related appendages, including members of the type IV filament super-family	26	341	Merger of TFF-SF v1.0 and TXSScan v1.0: - TFF-SF models from Denise et al. replace older model versions from TXSScan version 1.0 (Abby et al. ) for the T2SS, Tad and T4aP systems. Models added: ComM, T4bP and Archaeal_T4P. - Hierarchy of models by domain of life, and then by membrane type to allow to search only the relevant models.

382

383

**Table 2.** The *macsydata* companion tool to handle *macsy-model* packages

<i>macsydata</i> command	Description	Examples
<i>available</i>	List all <i>macsy-model</i> packages available at the default or specified organization	<i>macsydata available</i>
<i>list</i>	List installed packages	<i>macsydata list</i>
<i>check</i>	Allows to check the sanity and consistency of a <i>macsy-model</i> package before diffusion	<i>macsydata check</i>
<i>install / uninstall</i>	Automatically retrieve and install (uninstall) the designated package	<i>macsydata install</i> TFF-SF
<i>cite</i>	Displays the citation information stored in the metadata.yml file of the package	<i>macsydata cite</i> TFF-SF
<i>definition</i>	Displays the definition(s) (XML file) of the specified model(s). It can be a directory containing several models to display.	<i>macsydata definition</i> TXSS T1SS <i>macsydata definition</i> TXSS/archaeal <i>macsydata definition</i> --models-dir my-models System1
<i>search</i>	Search for the models based on their names, or based on string searches in the models' description	<i>macsydata search</i> TXSS <i>macsydata search</i> -s Secretion
<i>macsydata</i> <subcommand> --help	List the help message for the specified sub-command	<i>macsydata search</i> --help

384

385

386

387

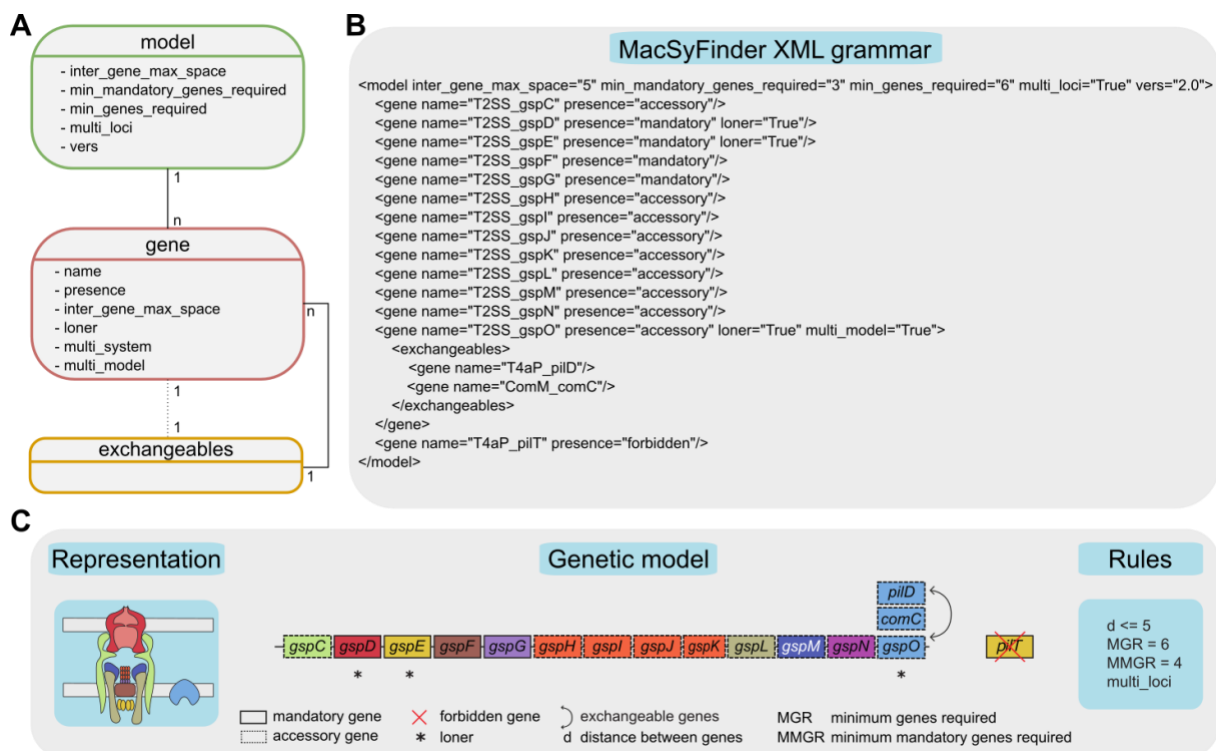
388

389

390

To illustrate the interest of the novel file architecture, we carried under the v2 grammar and assembled into a new version of “TXSScan” (v1.1.0) the models for the type IV filament super-family (“TFF-SF”) and for the protein secretion systems (former “TXSScan”, v1.0.0) that we had previously developed for MacSyFinder v1 (Abby et al., 2016; Denise et al., 2019). These systems represent a coherent set of bacterial appendages dedicated to motility and secretion that share evolutionary relations (see

391 (Denise et al., 2020) for a review). We organized the models into sub-directories with  
 392 respect to relevant biological criteria (Fig. S1). The models' sub-directories were split  
 393 by domains of life (archaea versus bacteria) and then by membrane type (monoderm  
 394 bacteria versus diderm bacteria). This new architecture enables the search for all  
 395 models at once, or only the domain-specific ones or those specific to a given bacterial  
 396 membrane type. This allows more targeted and less costly searches for biological  
 397 systems. Of note, the XML grammar simplifications introduced in v2 enabled the  
 398 production of much more compact, readable, and simple models (Fig. 3). For example,  
 399 the definition of the type III secretion system (T3SS) now consists of 20 lines for 15  
 400 listed components, whereas the v1 version counted 52 lines. v2 versions of the popular  
 401 TXSScan and TFF-SF models are also available as originally published (Abby et al.,  
 402 2016; Denise et al., 2019) (v1.0.0 versions of TXSScan and TFFscan respectively,  
 403 Table 1). Yet the new search engine has a different behaviour than v1; it will produce  
 404 different results in some cases (see below and Fig. 4).  
 405



406  
407

408 **Figure 3. Description of the hierarchical grammar used in MacSyFinder models**  
 409 **and example of the T2SS model.**

410 **A.** The “model” is the root element of the XML document according to the grammar. It  
 411 represents the system to model and contains at least one element “gene”. The “gene”  
 412 element describes the components constituting a system. It may contain one element  
 413 “exchangeables”. The dashed line between “gene” and “exchangeables” illustrate the  
 414 fact that a gene does not necessary contain an “exchangeables” element. The  
 415 “exchangeables” element contains a set of genes (one at least) that can replace  
 416 functionally the parent “gene” in the system quorum. The one-to-many relationships  
 417 between the different elements is represented by lines connecting the boxes, with the  
 418 cardinality of the relationship appearing next to the element. The diverse possible  
 419 features of each element are represented in the corresponding boxes. **B.** XML



420 grammar of the T2SS from TXSScan v1.1.0 (and TFFscan v1.0.0) (Abby et al., 2016;  
421 Denise et al., 2019). **C.** A schematic representation of the T2SS machinery spanning  
422 the membranes of a diderm bacterium is displayed on the left. The genetic model  
423 corresponding to the T2SS model in panel B is illustrated in the central part, with gene  
424 boxes filled with the colour of the corresponding proteins on the T2SS schema. The  
425 quorum and co-localization rules to fulfil the T2SS model are described on the right.  
426 Gene components' names were abbreviated in the genetic model compared to the  
427 names in the XML model. The C panel was derived from (Denise et al., 2019).

428

### 429 **II/ A new system modelling and search engine for a more relevant exploration of** 430 **possible systems**

431 MacSyFinder v1 had a greedy search engine with sub-optimal and sometimes  
432 unexpected behaviours, especially in complex cases such as co-localized systems or  
433 those with multiple occurrences in a genome. The novel v2 search engine explores the  
434 space of possible solutions more thoroughly. It provides optimal solutions with an  
435 explicit scoring system favouring complete but concise systems (Fig. 1). A fundamental  
436 improvement is that the systems are now searched one by one: the identified  
437 components are filtered by type of system and assembled in clusters if relevant (Fig.  
438 1). The new search engine can thus resolve more complex case scenarios than the  
439 previous one (see below and Fig. 4). Using a system-by-system approach also  
440 prevents the spurious elimination of relevant candidate systems, e.g., when a  
441 component from another system is within a cluster of the candidate system. This was  
442 a cause for the elimination of certain valid systems in v1 (see below and Fig. 4).

443

444 The combinatorial exam of sets of components and clusters to build up candidate  
445 systems (see Materials and Methods) allows to deal with more complex cases, e.g. the  
446 occurrence of multiple scattered systems (see below and Fig. 4 for an example).  
447 However, the combinatorial exploration may be computationally costly, especially  
448 when there are many occurrences of clusters and components. This cost is partly  
449 relieved by filtering the components using the GA scores (or other criteria), because  
450 this effectively removes many false positives and leaves fewer components and  
451 clusters to consider. Yet when testing the new search engine, we were sometimes  
452 confronted to cases of genomes with dozens of hits for “out-of-cluster” components  
453 (loner or multi\_system components). The analysis of all combinations of such  
454 components can be extremely costly. To make these cases manageable, MacSyFinder  
455 v2 uses a heuristic that considers several occurrences of the same “out-of-cluster”  
456 component as a single component when forming combinations of potential systems.  
457 This “representative” is selected as the best matching component (best HMMER  
458 score). The other “out-of-cluster” components detected are kept and listed separately  
459 in dedicated files (best\_solution\_loners.tsv and best\_solution\_multisystems.tsv). This  
460 makes the combinatorial exploration of solutions manageable in most if not all the  
461 cases. If this is not the case, we advise the users to revise their system's modelling  
462 strategy and/or HMM profiles specificity (e.g., increase the GA score thresholds).  
463 Finally, the graph-based search for the best solution (see Materials and Methods and  
464 Fig. 2B) also provides sub-optimal solutions that may be of interest to the user in some  
465 situations (system variants discovery, detection of degraded systems, etc.).

466

467 We illustrate and discuss in the following sections the advantage of MacSyFinder v2

468 over v1 for the search of molecular systems. For this, we present its application to three  
469 types of systems: the CRISPR-Cas system (CasFinder package), the conjugative  
470 systems (CONJscan package), and the type IV filament super-family (TFFscan and  
471 TXSScan packages).

472

### 473 **III/ Application of MacSyFinder v2 to CasFinder**

474

475 CRISPR-Cas systems are adaptive immune systems that protect bacteria and archaea  
476 from invasive agents (phages, plasmids...) (Hampton et al., 2020). A typical CRISPR-  
477 Cas system consists of a CRISPR array and adjacent cluster of *cas* (CRISPR  
478 associated) genes that form one or more operons of 1 to 13 genes (Fig. 4A) (Makarova  
479 et al., 2020). As CRISPR arrays do not code for proteins, this part of the system is not  
480 identified by MacSyFinder. The *cas* genes clusters are very diverse and are currently  
481 classified into two classes, six types (I-VI) and more than 30 subtypes based on their  
482 composition in *cas* genes (Makarova et al., 2020). We have previously developed a  
483 package of models called CasFinder dedicated to the detection of CRISPR-Cas  
484 systems using MacSyFinder v1 (Abby et al., 2014; Couvin et al., 2018). We hereby  
485 propose an updated and improved version of CasFinder that benefits from the new  
486 features of MacSyFinder v2.

487

#### 488 **The graph-based approach improves tandem systems detection**

489 CRISPR-Cas systems can be subdivided into three distinct, though partially  
490 overlapping, functional modules. Some of these, e.g., the adaptation module mainly  
491 composed of Cas1, Cas2, and Cas4 proteins, may be very similar between subtypes  
492 or even types, making the detection of tandem systems particularly challenging. With  
493 the v2 new search engine, all systems are searched one by one. The best possible  
494 combination of systems is retrieved using a graph-based approach, which significantly  
495 improves the identification of tandem systems. This improvement is even more  
496 important when the number of tandem systems exceeds two, as MacSyFinder v1 could  
497 not handle these rare complex situations at the subtype level (Fig. 4B).

498

#### 499 **The “Multi\_model” component feature enables tandem, overlapping systems 500 detection**

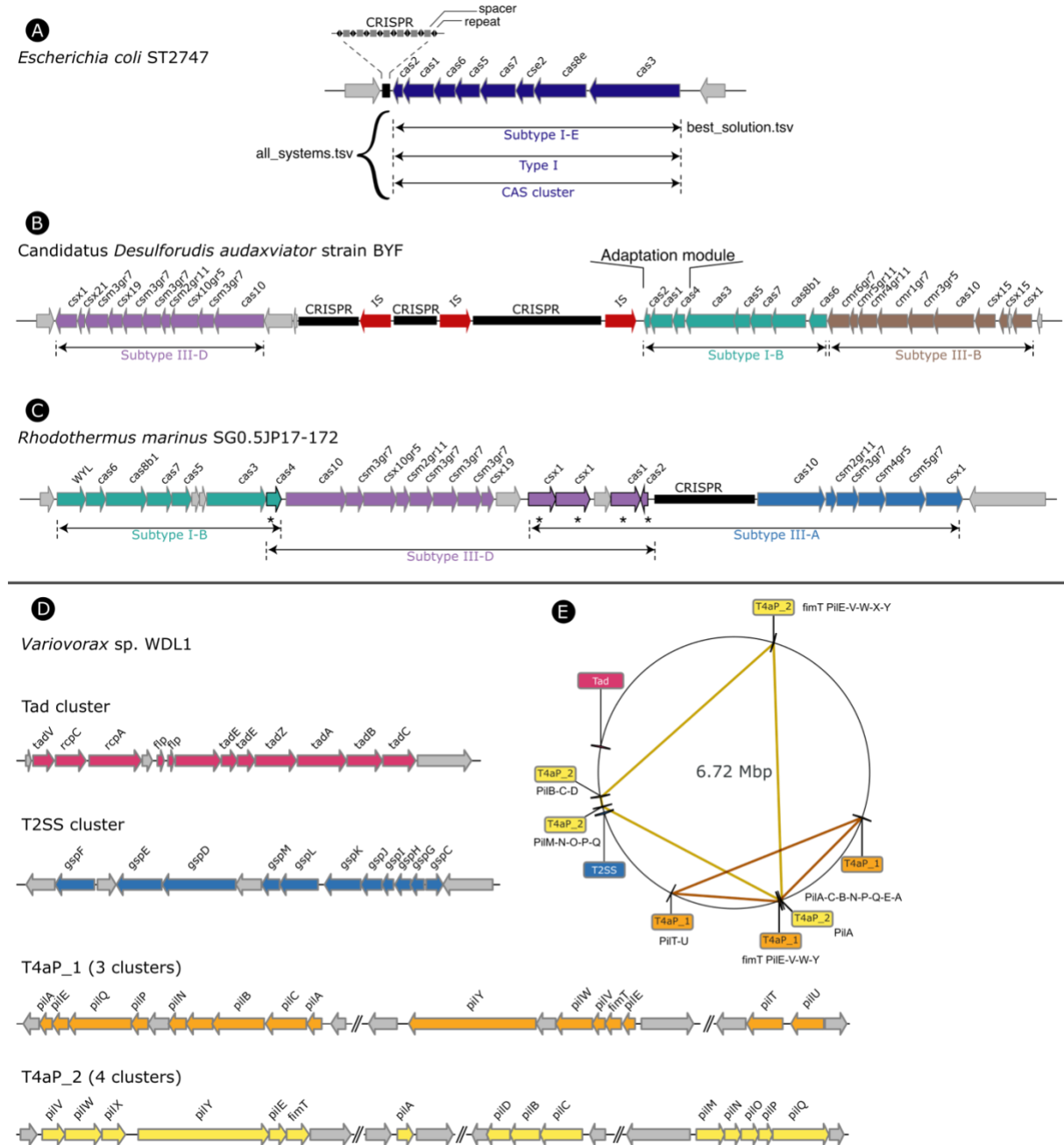
501 Most CRISPR-Cas systems have an adaptation module when they are alone. But,  
502 when they are in tandem, it is not uncommon to find that only one module is present  
503 for both systems (Bernheim et al., 2019). This complicates its detection, especially  
504 when it is located between tandem systems. In v1, the adaptation module was  
505 assigned to one of the two systems at the risk of missing the second one if the latter  
506 turned out to be too small (i.e., with a minimum number of required genes lower than  
507 the defined threshold). In v2, thanks to the new “multi\_model” gene feature, it is  
508 possible to allow a component to be present in different models. Thus, by defining the  
509 proteins involved in the adaptation module as “multi\_model”, they are assigned to the  
510 two overlapping systems (Fig. 4C).

#### 511 **The new search engine and scoring system allow searching for different levels 512 of classification simultaneously**

513 Some Cas subtypes are extremely similar in terms of gene content and require very  
514 precise decision rules to distinguish them. However, the more precise these rules are,  
515 the higher the risk is of missing systems. To overcome this difficulty, we have previously

516 defined different sets of models providing detection at three levels of classification,  
517 from the most permissive to the most specific one: (1) a general model (called  
518 CAS\_cluster) allowing the identification of any cluster of *cas* genes, (2) a set of models  
519 for detection at the type level, (3) and finally a set of models for detection at the subtype  
520 level. MacSyFinder v1 analyzed the models one by one in a pre-defined order and  
521 selected the first model whose rules were satisfied. Using all three sets of models  
522 simultaneously meant that not all possibilities were explored. Thanks to the new v2  
523 search engine and scoring system, all models can now be analyzed at once. Therefore,  
524 it is possible that the same cluster is detected at several classification levels (Fig. 4A).  
525 The choice of the best solution presented to the user among these different  
526 assignments is based on the score of each candidate, then on their wholeness  
527 (proportion of genes found over the number of listed ones, or over “max\_nb\_genes” if  
528 defined in the model). Here, the subtype level models have been defined with a  
529 “max\_nb\_genes” parameter lower than for the models higher in the classification.  
530 Thus, for a given system that will obtain the same score from several classification  
531 levels, the most specific one will obtain the higher system’s wholeness, ensuring the  
532 most specific annotation is proposed as the best solution. We thus favoured annotation  
533 at the subtype level as being by far the most informative. Still, when the subtype-level  
534 search fails, the program allows the detection of atypical or degenerated clusters via  
535 the models at the other levels (type-level or general case).

536



537

538

539

540

541

542

543

544

545

546

547

548

549

550

**Figure 4. Application of MacSyFinder v2 to CasFinder (v3.1.0) and TFFscan (v1.0.0).** (A) CRISPR-Cas system has two parts: a CRISPR array and a cluster of *cas* genes. The new MacSyFinder search engine simultaneously annotates Cas clusters at 3 levels of classification from the most accurate (i.e. the subtype level) to the most permissive. When possible, it favours as the best solution the annotation at the subtype level but allows to recover atypical or degenerated systems with the 2 other levels of classification. (B) The combinatorial approach for the search of the best solution improves the detection of tandem systems. All models are tested and challenged, then the best combination of systems is determined. Here, it reveals the presence of 3 systems of different subtype in tandem (one color per subtype). (C) The new search engine avoids overlap between different candidate systems to determine the best solution(s), unless specified in the model with the `multi_system` or `multi_model` features. As illustrated, the adaptation module (*cas1*, *cas2* and *cas4*) has been defined

551 as “multi\_model” (indicated by a star\*) in some subtype models and can thus be  
552 assigned to 2 systems in tandem, which improves their identification. Without this  
553 feature newly implemented in v2, one of the two systems would be lost. **(D)** Several  
554 members of the Type IV filament super-family (TFF-SF) could be found in the genome  
555 of *Variovorax* sp. WDL1. The new search engine enables the annotation of two distinct  
556 T4aP in the same genome. Here we can observe that the two detected T4aP are  
557 gathering clusters of different and complementary gene composition, underlying their  
558 coherence. The two strokes between each gene cluster signifies that the clusters are  
559 not close to each other on the chromosome. **(E)** The location of the T4aP gene clusters  
560 is displayed along the circular chromosome. A polygon connects the different parts of  
561 a same system with colors matching that of the systems in panel D. In all panels, genes  
562 are represented by arrows, their length indicates the gene length, and their direction  
563 indicates the gene orientation.

564

#### 565 **IV/ Application of MacSyFinder v2 to TXSScan and CONJscan**

566 **The new search engine and scoring system allow the retrieval of various**  
567 **occurrences of Type IV pili encoded at multiple loci.**

568 The type IV filaments super-family (TFF-SF) is a family of homologous machineries  
569 involved in bacterial and archaeal motility (e. g., the type IVa pilus “T4aP” and archaeal  
570 flagellum), toxin secretion (e.g., type II secretion systems, T2SS) or exogenous DNA  
571 acquisition (e. g., competence apparatus, Com) (Pelicic, 2008). Some members of the  
572 TFF-SF have their genes scattered across the genome (e.g., T4aP and some T2SS),  
573 and some genomes may harbour several scattered occurrences of the same system  
574 (Denise et al., 2019). In this case, it is not trivial to identify and discriminate the  
575 occurrences of the different systems. The search engine of MacSyFinder v 1 collected  
576 occurrences of the same system as one large system containing multiple copies of  
577 several genes. The new v2 search engine examines and then scores all possible  
578 combinations of gene clusters and (authorized) out-of-cluster genes eligible as  
579 systems. The scoring of these candidate systems penalizes the presence of the same  
580 gene in several gene clusters. This approach favours solutions presenting complete  
581 yet concise systems. For example, it allows the separation of two different multi-loci  
582 T4aP found in the same genome (Fig. 4D-E).

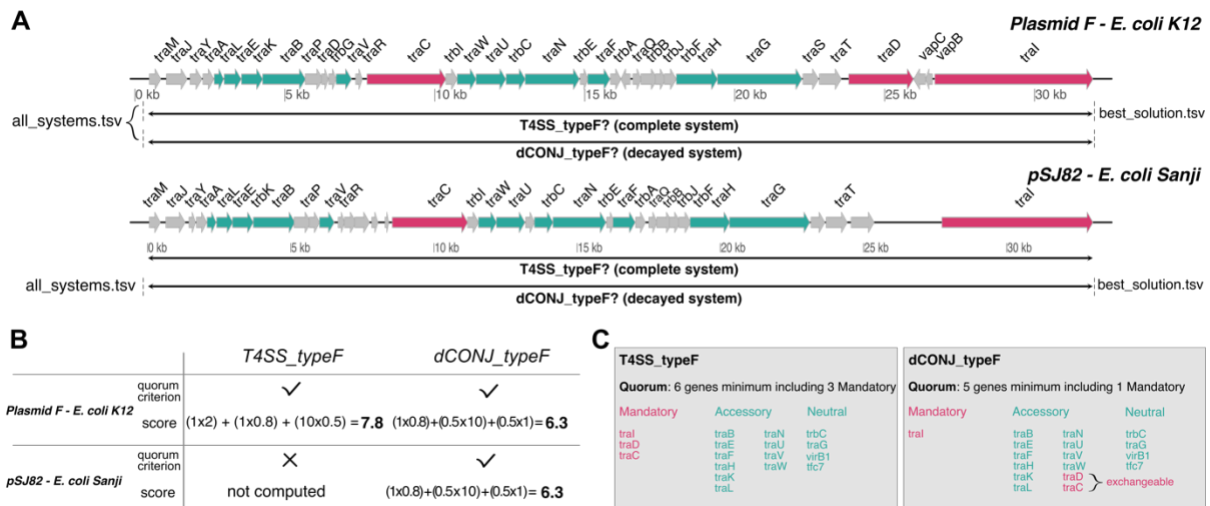
583 **The new scoring scheme allows to distinguish putatively degenerate**  
584 **conjugative elements from the others.**

585 Integrative conjugative elements and conjugative plasmids are very abundant mobile  
586 genetic elements that can transfer themselves from one bacterium to another. To do  
587 so, they encode a conjugative system that includes a relaxase (MOB) and a mating  
588 pair formation (MPF) machinery responsible for pilus biogenesis and mating junctions  
589 (de la Cruz et al., 2010). The known relaxases are currently searched using 11 HMM  
590 profiles, and the MPFs are classified into eight different types (FA, FATA, B, C, F, G,  
591 I, and T). Together, they make for eight models of T4SS (Guglielmini et al., 2013).  
592 MPFs include numerous genes, between eight to several dozens. However, the  
593 conserved set of genes seen as mandatory is much smaller (relaxase, VirB4, coupling  
594 protein), and the other conserved components oscillate between seven and 27. From  
595 wet-lab experiments to pandemic studies or phylogenetic analyses, discriminating  
596 between complete transferrable elements and incomplete, potentially immobile  
597 conjugative elements, is crucial. We have recently shown that degenerate conjugative

598 elements are not rare (Coluzzi et al., 2022). Hence, it would be important to have an  
 599 easy way to identify complete and incomplete systems. To tackle this problem, we  
 600 developed macsy-models taking advantage of the new scoring scheme implemented  
 601 in v2.

602 All systems can be tested and challenged at once in the new version. The selection of  
 603 the best solution among different candidate systems is based on the score of each  
 604 candidate. Using this new feature, we created models designed to compete with each  
 605 other (Fig. 5). For each conjugative system, one model was designed to detect  
 606 complete systems, while the other was designed to detect both complete and  
 607 incomplete systems. Used independently, the complete model would only detect  
 608 complete systems, and the incomplete model would indiscriminately detect complete  
 609 and incomplete systems. However, when used together in competition, the scoring  
 610 system attributes actual complete systems to the complete model while incomplete  
 611 systems are only detected by the “incomplete” model (Fig. 5).

612



613 **Figure 5. Application of MacSyFinder v2 to distinguish complete and incomplete**  
 614 **conjugative systems on bacterial plasmids with CONJscan**  
 615 **v2.0.1. A.** Representation of a complete conjugative system (top) and a decayed  
 616 conjugative system (bottom). Arrows represent the predicted genes of the plasmids  
 617 and their orientation. Mandatory and accessory components of the systems are  
 618 represented in fuchsia and cyan respectively. **B.** Description of the score for the  
 619 complete MPF<sub>F</sub> (T4SS\_TypeF) model and “decayed” MPF<sub>F</sub> model (dCONJ\_typeF)  
 620 when computed by the scoring scheme of MacSyFinder v2 (detailed in Fig.  
 621 2A). CONJScan plasmids’ models were used all at once with the “all” option.  
 622 **C:** Difference between the complete and decayed models for the MPF<sub>F</sub>. Both models  
 623 list the same components, but the required quorum of mandatory genes and total  
 624 genes required are different. The model designed to detect complete systems  
 625 (T4SS\_typeF) requires 3 mandatory genes and 6 genes minimum, while the “decayed”  
 626 model (dCONJ\_typeF) was designed to require only 1 mandatory gene and the other  
 627 mandatory genes were set as “accessory” and exchangeable between each other.  
 628 Thus, we ensure that if the quorum is reached for a complete system, the score of the  
 629 “complete” model is always higher than the score of the “decayed” model.  
 630

631

632 **CONCLUSION**

633 MacSyFinder leverages the power of comparative genomics for accurate system-level  
634 annotation of microbial genomes. MacSyFinder version 2 enables more relevant and  
635 comprehensive system modelling and search capacities. The variety of the  
636 applications illustrated here and elsewhere demonstrates the potential of MacSyFinder  
637 to annotate many other cellular functions, including biosynthetic gene clusters,  
638 metabolic and signalling pathways. The *macsydata* tool and “MacSy Models” Github  
639 organization allow systems’ modellers to easily share their macsy-model packages.  
640 We hope this will increase the visibility of their contribution and enhance the  
641 development of novel models for other molecular systems.

642

#### 643 **DATA AVAILABILITY**

644 MacSyFinder source code and the hereby presented macsy-model packages are  
645 available at the following Github repositories: [https://github.com/gem-](https://github.com/gem-pasteur/macsyfinder)  
646 [pasteur/macsyfinder](https://github.com/macsy-models) and <https://github.com/macsy-models>.

647

#### 648 **AUTHORS’ CONTRIBUTIONS**

649 BN, EPCR and SSA designed the new version of MacSyFinder. BN conceived the  
650 software architecture and design, the test design, and performed the implementation.  
651 RD, CC, MT, EPCR and SSA tested MacSyFinder. RD, CC, MT and SSA updated and  
652 distributed the presented macsy-model packages on the dedicated repository. RD, CC  
653 and MT analysed the results of MacSyFinder detection and implemented GA scores  
654 within HMM profiles of the presented macsy-model packages. EPCR and SSA wrote  
655 the first versions of the manuscript, and all authors contributed to and approved the  
656 final versions of the manuscript.

657

#### 658 **ACKNOWLEDGEMENTS**

659 The authors are grateful to Yoann Dufresne for the suggestion to use the NetworkX  
660 library to address the weighted maximum clique search problem. This work used the  
661 computational and storage services (TARS cluster) provided by the IT department at  
662 Institut Pasteur, Paris.

663

#### 664 **FUNDINGS**

665 EPCR lab acknowledges funding from the INCEPTION project (ANR-16-CONV-0005),  
666 Equipe FRM (Fondation pour la Recherche Médicale) : EQU201903007835, and  
667 Laboratoire d’Excellence IBEID Integrative Biology of Emerging Infectious Diseases  
668 (ANR-10-LABX-62-IBEID). SSA received financial support from the CNRS and TIMC  
669 lab (INSIS “starting grant”) and the French National Research Agency,  
670 “Investissements d’avenir” program ANR-15-IDEX-02.

671 **REFERENCES**

672

673 Abby, S. S., J. Cury, J. Guglielmini, B. Néron, M. Touchon, and E. P. C. Rocha.  
674 2016. Identification of protein secretion systems in bacterial genomes.  
675 *Scientific Reports* 6: 23080.

676 Abby, S. S., B. Neron, H. Menager, M. Touchon, and E. P. Rocha. 2014.  
677 MacSyFinder: a program to mine genomes for molecular systems with an  
678 application to CRISPR-Cas systems. *PLoS One* 9: e110726.

679 Abby, S. S., and E. P. C. Rocha. 2012. The Non-Flagellar Type III Secretion System  
680 Evolved from the Bacterial Flagellum and Diversified into Host-Cell Adapted  
681 Systems. *PLoS Genet* 8: e1002983.

682 Adam, P. S., G. Borrel, and S. Gribaldo. 2019. An archaeal origin of the Wood-  
683 Ljungdahl H4MPT branch and the emergence of bacterial methylophony. *Nat*  
684 *Microbiol* 4: 2155–2163.

685 Bernheim, A., D. Bikard, M. Touchon, and E. P. C. Rocha. 2019. Atypical  
686 organizations and epistatic interactions of CRISPRs and cas clusters in  
687 genomes and their mobile genetic elements. *Nucleic Acids Research*:  
688 gkz1091.

689 Brandes, U., and T. Erlebach. 2005. Network Analysis. Methodological Foundations.  
690 1st ed. U. Brandes, and T. Erlebach [eds.], Springer Berlin, Heidelberg.

691 Chibani, C. M., A. Mahnert, G. Borrel, A. Almeida, A. Werner, J. F. Brugere, S.  
692 Gribaldo, et al. 2022. A catalogue of 1,167 genomes from the human gut  
693 archaeome. *Nat Microbiol* 7: 48–61.

694 Coluzzi, C., M. P. Garcillan-Barcia, F. de la Cruz, and E. P. C. Rocha. 2022.  
695 Evolution of Plasmid Mobility: Origin and Fate of Conjugative and  
696 Nonconjugative Plasmids. *Mol Biol Evol* 39.

697 Couvin, D., A. Bernheim, C. Toffano-Nioche, M. Touchon, J. Michalik, B. Neron, E. P.  
698 C. Rocha, et al. 2018. CRISPRCasFinder, an update of CRISRFinder,  
699 includes a portable version, enhanced performance and integrates search for  
700 Cas proteins. *Nucleic Acids Res* 46: W246–W251.

701 de la Cruz, F., L. S. Frost, R. J. Meyer, and E. L. Zechner. 2010. Conjugative DNA  
702 metabolism in Gram-negative bacteria. *FEMS Microbiol Rev* 34: 18–40.

703 Cury, J., S. S. Abby, O. Doppelt-Azeroual, B. Néron, and E. P. C. Rocha. 2020.  
704 Identifying Conjugative Plasmids and Integrative Conjugative Elements with  
705 CONJscan. In F. de la Cruz [ed.], Horizontal Gene Transfer, 265–283.  
706 Springer US, New York, NY.

707 Cury, J., M. Touchon, and E. P. C. Rocha. 2017. Integrative and conjugative  
708 elements and their hosts: composition, distribution and organization. *Nucleic*  
709 *Acids Research* 45: 8943–8956.



- 710 Dandekar, T., B. Snel, M. Huynen, and P. Bork. 1998. Conservation of gene order: a  
711 fingerprint of proteins that physically interact. *Trends in Biochemical Sciences*  
712 23: 324–328.
- 713 Denise, R., S. S. Abby, and E. P. C. Rocha. 2019. Diversification of the type IV  
714 filament superfamily into machines for adhesion, protein secretion, DNA  
715 uptake, and motility. *PLOS Biology* 17: e3000390.
- 716 Denise, R., S. S. Abby, and E. P. C. Rocha. 2020. The Evolution of Protein Secretion  
717 Systems by Co-option and Tinkering of Cellular Machineries. *Trends in*  
718 *Microbiology* 28: 372–386.
- 719 Eddy, S. R. 2011. Accelerated Profile HMM Searches. *PLoS Comput Biol* 7:  
720 e1002195.
- 721 Guglielmini, J., F. de la Cruz, and E. P. C. Rocha. 2013. Evolution of conjugation and  
722 type IV secretion systems. *Molecular biology and evolution* 30: 315–331.
- 723 Haft, D. H., J. D. Selengut, and O. White. 2003. The TIGRFAMs database of protein  
724 families. *Nucleic Acids Research* 31: 371–373.
- 725 Hagberg, A. A., D. A. Schult, and P. J. Swart. 2008. Exploring Network Structure,  
726 Dynamics, and Function using NetworkX. In G. Varoquaux, T. Vaught, and J.  
727 Millman [eds.],.
- 728 Hampton, H. G., B. N. J. Watson, and P. C. Fineran. 2020. The arms race between  
729 bacteria and their phage foes. *Nature* 577: 327–336.
- 730 Huynen, M., B. Snel, W. Lathe, and P. Bork. 2000. Predicting protein function by  
731 genomic context: quantitative evaluation and qualitative inferences. *Genome*  
732 *Res* 10: 1204–10.
- 733 Makarova, K. S., Y. I. Wolf, J. Iranzo, S. A. Shmakov, O. S. Alkhnbashi, S. J. J.  
734 Brouns, E. Charpentier, et al. 2020. Evolutionary classification of CRISPR–  
735 Cas systems: a burst of class 2 and derived variants. *Nature Reviews*  
736 *Microbiology* 18: 67–83.
- 737 Pelicic, V. 2008. Type IV pili: e pluribus unum? *Molecular Microbiology* 68: 827–837.
- 738 Pende, N., A. Sogues, D. Megrian, A. Sartori-Rupp, P. England, H. Palabikyan, S. K.  
739 R. Rittmann, et al. 2021. SepF is the FtsZ anchor in archaea, with features of  
740 an ancestral cell division system. *Nat Commun* 12: 3214.
- 741 Rendueles, O., M. Garcia-Garcera, B. Neron, M. Touchon, and E. P. C. Rocha. 2017.  
742 Abundance and co-occurrence of extracellular capsules increase  
743 environmental breadth: Implications for the emergence of pathogens. *PLoS*  
744 *Pathog* 13: e1006525.
- 745 Sharp, C., and K. R. Foster. 2022. Host control and the evolution of cooperation in  
746 host microbiomes. *Nat Commun* 13: 3567.

- 747 Sonnhammer, E. L., S. R. Eddy, and R. Durbin. 1997. Pfam: a comprehensive  
748 database of protein domain families based on seed alignments. *Proteins* 28:  
749 405–420.
- 750 Taib, N., D. Megrian, J. Witwinowski, P. Adam, D. Poppleton, G. Borrel, C. Beloin,  
751 and S. Gribaldo. 2020. Genome-wide analysis of the Firmicutes illuminates the  
752 diderm/monoderm transition. *Nat Ecol Evol* 4: 1661–1672.
- 753 Teichmann, S. A., and M. M. Babu. 2002. Conservation of gene co-regulation in  
754 prokaryotes and eukaryotes. *Trends Biotechnol* 20: 407–10; discussion 410.
- 755 Tesson, F., A. Herve, E. Mordret, M. Touchon, C. d’Humieres, J. Cury, and A.  
756 Bernheim. 2022. Systematic and quantitative view of the antiviral arsenal of  
757 prokaryotes. *Nat Commun* 13: 2561.
- 758 Vallenet, D., A. Calteau, M. Dubois, P. Amours, A. Bazin, M. Beuvin, L. Burlot, et al.  
759 2020. MicroScope: an integrated platform for the annotation and exploration of  
760 microbial gene functions through genomic, pangenomic and metabolic  
761 comparative analysis. *Nucleic Acids Res* 48: D579–D589.
- 762