MacSyFinder v2: An improved search engine to model and identify molecular systems in genomes

Bertrand Néron, Rémi Denise, Charles Coluzzi, Marie Touchon, Eduardo Rocha, Sophie Abby

MacSyFinder v2: An improved search engine to model and identify molecular systems in genomes

Bertrand Néron[1], Rémi Denise[2], Charles Coluzzi[2], Marie Touchon[2], Eduardo Rocha[2], Sophie Abby[3]

[1]Institut Pasteur, Université de Paris Cité, Bioinformatics and Biostatistics Hub, 75015 Paris, France
[2]Institut Pasteur, Université de Paris Cité, CNRS UMR3525, Microbial Evolutionary Genomics, 75015 Paris, France
[3]CNRS, TIMC-IMAG, GEM team, Université Grenoble Alpes, La Tronche

bneron@pasteur.fr
sophie.abby@univ-grenoble-alpes.fr

Complex cellular functions are most often encoded by a set of genes rather than individual ones. Furthermore, the genes in such "systems" are often encoded nearby in microbial genomes.

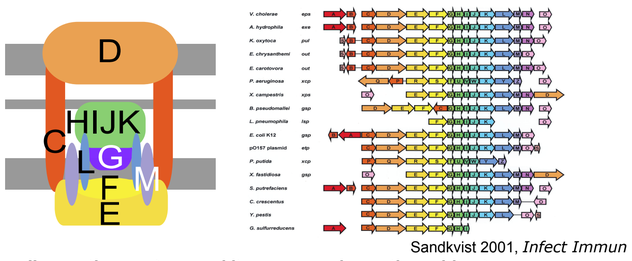
MacSyFinder uses these properties to model and then accurately annotate cellular functions in microbial genomes at the system-level rather than at the individual-gene level.

We present a major release of MacSyFinder, MacSyFinder v2. Among other new features we introduce a more intuitive and comprehensive search engine to identify all the best candidate systems and sub-optimal ones.

We also present the novel *macsydata* companion tool that enables the easy installation and broad distribution of the models developed for MacSyFinder (macsy-models) from GitHub repositories.

Principles of MacSyFinder

Microbial machineries and pathways (hereafter called "systems") can be very complex and involve many proteins. In the genomes of Bacteria and Archaea, the components of these systems are often encoded in a **highly organized and conserved** way, involving one or a few operons with functionally-related genes. Using the evolutionary conserved properties of these systems can ensure their **highly specific annotations**.



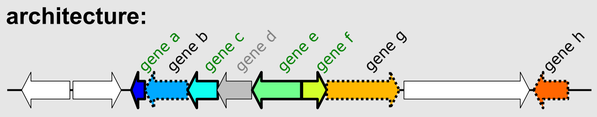
Sandkvist 2001, *Infect Immun*

The type II secretion system, and its conserved genetic architecture.
This cellular machinerie enables the release of degradation enzymes and toxins such as the cholera toxin. It is encoded by more than a dozen of genes in a conserved locus in the genomes of many bacteria including *Legionella*, *Vibrio*, *Pseudomonas*, etc...

We created a **framework to model the conserved properties** of molecular systems for their subsequent annotation:
- their **set of genes** (components) and quorum for a full system
- their **genomic architecture**

Once modelled by the user in XML model files (see below), the MacSyFinder search engine seeks the listed system's components by sequence similarity (HMMER, *hmmer.org*), and then screen the **genomic architecture and content** of matched components to assess the **presence or absence** of a complete system.

Genomic architecture:



Quorum rules: min-number-mandatory-genes = 4
min-number-genes = 7

a c e f ☐ mandatory components
b g h ☐ accessory components
d neutral component
forbidden component

`full system A`

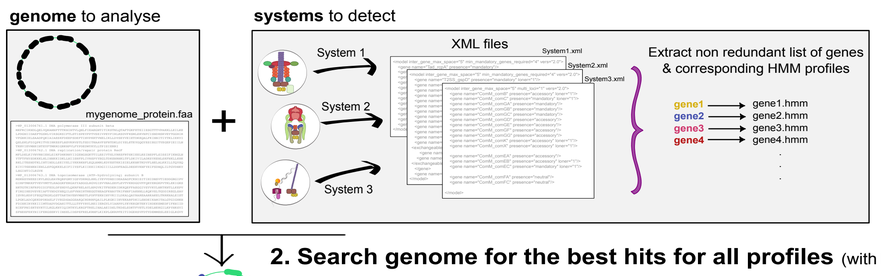
System search summary:

SeqID	Length	hit	System	SystemID	Role	Score	i-evalue
a	64	c1	systemA	systemA_1	mandatory	83	1.10^{-9}
b	119	c6	systemA	systemA_1	accessory	197	3.10^{-12}
c	75	c2	systemA	systemA_1	mandatory	92	8.10^{-10}
d	80	c10	systemA	systemA_1	neutral	88	5.10^{-7}

The new search engine and scoring system

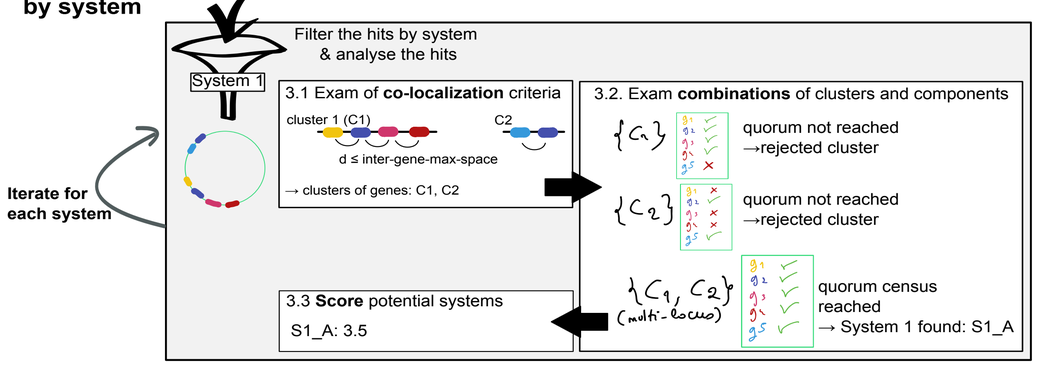
MacSyFinder v1 had a greedy search engine with sub-optimal behaviors, especially in complex cases such as co-localized systems or those with multiple occurrences in a genome. The novel v2 search engine explores the **space of possible solutions more thoroughly** and provides optimal solutions with an explicit **scoring system** favoring complete but concise systems. Here is an **overview of its functioning**:

1. Input files

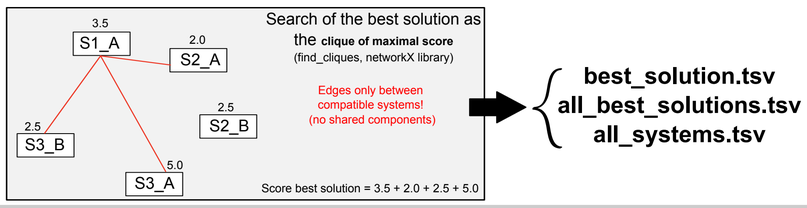


2. Search genome for the best hits for all profiles (with HMMER)

3. Search system by system



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



→ **best_solution.tsv**
all_best_solutions.tsv
all_systems.tsv

macsydata and the MacSy-models GitHub organization

In v2, the files required for MacSyFinder system search have been organized on the form of a macsy-model package. Model definitions are in the ***definitions*** directory (can contain sub-directories) and the HMM profiles in the ***profiles*** directory. The package must also contain a file with metadata describing the package (***metadata.yml***), and ***README*** and ***LICENSE*** files are recommended. It is also possible to ship specific default values for the model(s) in a configuration file (**model_conf.xml**). To publish and distribute your macsy-models, (1) ask for a repository at the **MacSy Models GitHub** organization, (2) tag and push the data. The package will then be automatically available to all users through the ***macsydata*** tool for browsing, installation and use with MacSyFinder v2.

Macsy-models package structure:

```
family_name
    |_______ metadata.yml
    |_______ LICENSE
    |_______ README.md
    |_______ model_conf.xml
    |_______ definitions
    |            |_______ model_1.xml
    |            |_______ model_2.xml
    |            :
    |_______ profiles
                 |_______ geneA.hmm
                 |_______ geneB.hmm
```

metadata.yml

```
---
maintainer:
  name: The name of the person who maintains/to contact
  email: The email of the maintainer (required)
short_desc: A one line description of the package (can
vers: The package version (required)
cite: The publication(s) to cite by the user when the p
doc: Where to find extended documentation (optional)
license: The license under the package is released (opt
copyright: The copyright of the package (optional)
```

If the package contains sub-families:

```
family_name
    |_______ metadata.yml
    |_______ LICENSE
    |_______ README.md
    |_______ model_conf.xml
    |_______ definitions
    |            |_______ subfamilyA
    |            |            |_______ model_1.xml
    |            |            |_______ model_2.xml
    |            |
    |            |_______ subfamilyB
    |            |            |_______ model_3.xml
    |            |            |_______ model_4.xml
    |            |
    |            :
    |
    |_______ profiles
                 |_______ geneA.hmm
                 |_______ geneB.hmm
```

model_conf.xml

```
<model_config>
    <weights>
        <mandatory>1</mandatory>
        <accessory>0.5</accessory>
        <exchangeable>0.8</exchangeable>
        <redundancy_penalty>1.5</redundancy_penalty>
        <out_of_cluster>0.7</out_of_cluster>
    </weights>
    <filtering>
        <e_value_search>0.1</e_value_search>
        <i_evalue_sel>0.01</i_evalue_sel>
        <coverage_profile>0.5</coverage_profile>
        <cut_ga>True</cut_ga>
    </filtering>
</model_config>
```

Contributing your macsy-model package:

```
>macsydata check my_models
>git tag <version>
>git push origin <version>
```

Listing macsy-models on GitHub:

`bioinfo@linux64:~$macsydata available`

- **CASFinder** detection of CRISPR-Cas systems
- **CONJScan** detection of conjugative systems
- **TFF-SF** detection of the type IV-filament super-family systems (T4P, Com...)
- **TXSS** detection of bacterial protein secretion systems and related appendages

Searching and installing macsy-models:

`bioinfo@linux64:~$macsydata search TXSS`
TXSS (1.0rc2) - TXSScan - Models for 15 types of bacterial protein secretion systems

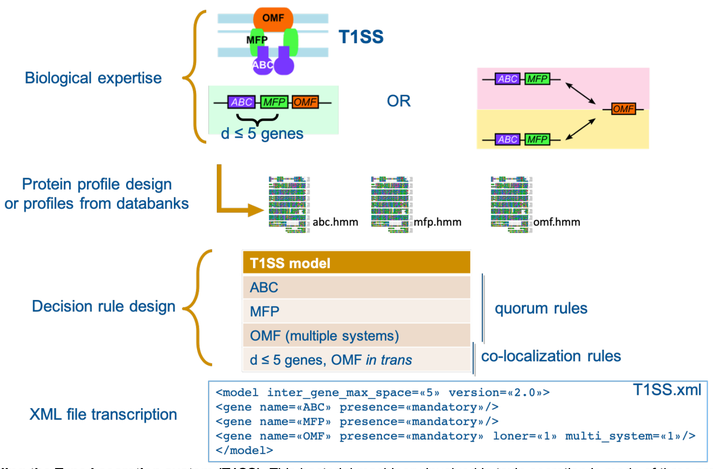
`bioinfo@linux64:~$macsydata search -S CRISPR`
CasFinder (3.1.0) - CasFinder - Models for detection of CRISPR-Cas systems.

`bioinfo@linux64:~$macsydata install -u CasFinder==3.1.0`
Downloading CasFinder (3.1.0).
Extracting CasFinder (3.1.0).
Installing CasFinder (3.1.0) in /Users/bioinfo/.macsyfinder/models
The models CasFinder (3.1.0) have been installed successfully.

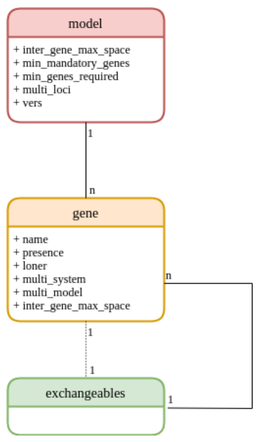
`bioinfo@linux64:~$macsyfinder --models CasFinder all --sequence-db ...`

MacSyFinder: how to create your own models?

The rationale behind MacSyFinder is to **translate any prior knowledge** of the molecular system to annotate into a model with a specific XML grammar. Relevant information pertains to the **set of conserved components** of the system and to their **genomic architecture**. HMM protein profiles that allow the similarity search of the systems' components can either be collected from databases (PFAM, TIGRFAM...) or designed by the user.



Modelling the Type I secretion system (T1SS). This bacterial machinery involved in toxin secretion is made of three proteins for which we have designed HMM profiles. The corresponding genes adopt two kinds of genomic architecture. They are either encoded next to each other (we set as 5 the maximal number of genes allowed between the genes to judge them contiguous), or only two of the genes (*abc* and *mfp*) co-localize, whenever the *omf* gene can be found elsewhere on the genome ("loner" gene feature). Moreover, *omf* can serve for several systems, and this corresponds to the "multi_system" gene feature. Once the HMM profiles are collected, and the content and co-localization rules identified and translated into an XML file, MacSyFinder can be ran for T1SS annotation in genomes.

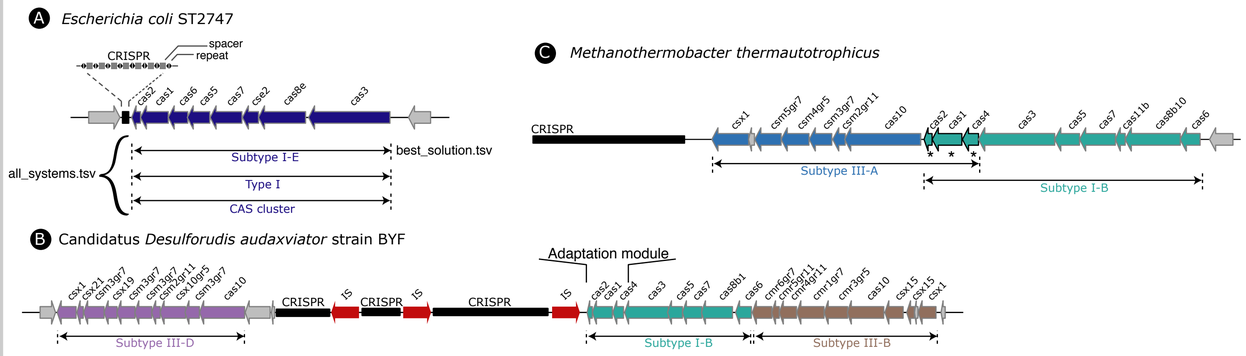


The XML grammar and its hierarchical structure for MacSyFinder v2 models. A model can list several genes, and genes can enclose "exchangeable" genes. The different possible features are listed. It has to be noted that the gene-defined features overcome the model-level ones. See the MacSyFinder User & Modeller guides for more details.

Whenever prior knowledge is scarce, MacSyFinder can also contribute to refine knowledge of a given system, as it is easily possible to change the parameters of the search in an iterative manner, before fixating relevant values for the model.

Application to CRISPR-Cas systems with the CasFinder package

CRISPR-Cas systems are adaptive immune systems that protect bacteria and archaea from invasive agents (phages, plasmids...). A typical CRISPR-Cas system consists of a CRISPR array and an adjacent cluster of cas (CRISPR associated) genes. As CRISPR arrays do not code for proteins, this part of the system is not identified by MacSyFinder. The cas genes clusters are very diverse and are currently classified into two classes, six types (I-VI) and more than 30 subtypes. This evolutionary classification is based on the identity, co-occurrence and composition of the cas genes in CRISPR–Cas loci, which makes CRISPR-Cas systems suitable for identification and typing by MacSyFinder.



The CasFinder 3.1.0 package (A) The new MacSyFinder search engine provides annotation of Cas clusters at 3 levels of classification from the most accurate (i.e. the subtype level) to the most permissive at once. CasFinder favors when possible 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 best solution search of v2 improves tandem systems detection. All models are tested and challenged, then the best combination of systems is determined revealing, here, the presence of 3 subtypes 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 as "multi_model" (indicated by a star*) in some subtype models and can thus be assigned to 2 systems in tandem, which improves their identification. Without this feature newly implemented in v2, one of the two systems would be lost.

References:
- Abby SS et al. MacSyFinder: A Program to Mine Genomes for Molecular Systems with an Application to CRISPR-Cas Systems. *Plos ONE*, 2014.
- Néron B et al. MacSyFinder v2: Improved features and search engine to model and identify molecular systems in genomes. *In preparation.*
- Macsy-Models: TXSScan → Abby et al. *Scientific Reports*, 2016 | TFF-SF → Denise et al. *Plos Biology*, 2019
CONJScan → Cury et al. *Methods Mol Biol,* 2020 | CasFinder → Couvin et al. *Nucleic Acids Research,* 2018

https://github.com/gem-pasteur/macsyfinder https://github.com/macsy-models

Distributed under GPL v3 licence. Requirement: Python (>= 3.7), HMMER suite (>=3.1b2). Python libraries: pyyaml, packaging, colorlog, pandas, networkX.