RNAcentral Help

RNAcentral Team

Nov 12, 2026

ABOUT RNACENTRAL

1 What do you want to do? 3

2 Funding 5

Documentation for RNAcentral, the open biodata resource for non-coding RNA sequences. These pages cover searching and browsing the data, accessing it programmatically or in bulk, and the annotations RNAcentral computes and imports.

New here? Start with *About RNAcentral* or the *Frequently Asked Questions*.

CHAPTER

ONE

WHAT DO YOU WANT TO DO?

I want to...	Go to
Find an RNA by name, gene or identifier	*Text search*
Find RNAs similar to my sequence	*Sequence search*
Understand an RNAcentral identifier	*RNAcentral identifiers*
Download data in bulk	*Bulk downloads*
Access the data from my own code	*REST API*, *Public Postgres database*, *MCP server*, Hugging Face
See where an RNA is in the genome	*Genome locations*
See the secondary structure of an RNA	*Secondary structure*
Find publications about an RNA	*LitScan*, *LitSumm*, *GOFlowLLM*
Link to RNAcentral from my website	*Linking to RNAcentral*
Cite RNAcentral	*Publications*
Get help	*FAQ*, *Contact us*

RNAcentral data are freely available under the *CC0 licence*.

CHAPTER

TWO

FUNDING

RNAcentral is supported by the BBSRC and Wellcome, and is developed at the EMBL-EBI.

2.1 About RNAcentral



RNAcentral is a free, public resource that offers integrated access to a **comprehensive** and **up-to-date** set of non-coding RNA sequences provided by a collaborating group of *Expert Databases* representing a **broad range of organisms and RNA types**.

The development of RNAcentral is coordinated by the European Bioinformatics Institute (EMBL-EBI), and RNAcentral is supported by the BBSRC and Wellcome.

To see RNAcentral in action, watch the *recorded webinar* on the training page.

2.1.1 What RNAcentral does

- **Data integration:** ncRNA sequences are imported from *expert databases* and made searchable through *text search* and *sequence similarity search*, with *bulk downloads* and *programmatic access*.
- **Stable identifiers:** every distinct sequence gets a *URS identifier*, with species-specific forms for referring to a sequence in a particular organism.
- **Genomic mapping:** sequences are mapped to reference genomes from more than 250 species and can be viewed in the *genome browser*.
- **Functional annotations:** *Rfam-based quality control*, *Gene Ontology annotations*, *RNA-target interactions*, *sequence features*, and *secondary structure diagrams*.
- **Literature linkage:** *LitScan* connects sequences to open access articles in Europe PMC, and *LitSumm* generates summaries of that literature.

2.1.2 Database growth

RNAcentral has grown from 7.7 million sequences and 8 expert databases at the 1.0beta release in 2014 to tens of millions of sequences from dozens of databases today. Hover over a point to see the figures for that release.

Switch between the two series with the tabs above the chart. See *Releases* for the release schedule and archived snapshots.

2.1.3 Timeline

1 July 2010: First RNAcentral workshop
Held at the Wellcome Trust Genome Campus (Hinxton, UK) the workshop formed the vision for RNAcentral.

22 September 2011: Publication in RNA
The paper outlining the need for RNAcentral brought together researchers from around the world.

1 October 2012: BBSRC grant
The development of RNAcentral is supported by a grant from Biotechnology and Biological Sciences Research Council (BBSRC). Award details

8 July 2013: First RNAcentral data hits public domain
The first RNAcentral dataset was imported into the European Nucleotide Archive. Release announcement

11 June 2014: Release 1.0beta
Release 1.0beta had 8 Expert Databases, including ENA, Rfam, miRBase, Vega, gtRNAdb, tmRNA Website, SRPDB, and lncRNAdb. Release announcement

10 September 2014: Release 1
The first public release saw RefSeq and RDP added to RNAcentral bringing the total number of Expert Databases to 10. Release announcement

28 October 2014: NAR Database Issue paper published
A new paper describing the first version of RNAcentral has been published in Nucleic Acids Research.

26 February 2015: Release 2
Release 2 brought in PDBe and snOPY as new Expert Databases and introduced a host of new website features (search results export, species-specific ids, and multiple search improvements) Release announcement

4 June 2015: Release 3
Three new expert databases were added to RNAcentral in release 3, including SGD, TAIR, and WormBase, as well as new sequence search powered by nhmmer Release announcement

27 November 2015: Release 4
Seven new expert databases were integrated into RNAcentral as well as nucleotide modifications from PDB and Modomics Release announcement

24 March 2016: Release 5
Release 5 saw a closer integration with Rfam and major improvements to search results Release announcement

1 July 2016: BBSRC grant
The development of RNAcentral is supported by a grant from Biotechnology and Biological Sciences Research Council (BBSRC). Award details

28 October 2016: NAR paper published
A new RNAcentral paper has been published in Nucleic Acids Research.

6 January 2017: Release 6
RNAcentral started linking to human non-coding RNAs from the HGNC database Release announcement

1 April 2017: Release 7
RNAcentral now includes data from Ensembl, GENCODE, and FlyBase Release announcement

10 November 2017: Release 8
RNAcentral now includes secondary structures and new data from GtRNAdb 2.1 and links to MGI Release announcement

11 April 2018: Release 9
RNAcentral now includes data from the Rat Genome Database Release announcement

20 August 2018: Release 10
RNAcentral now includes 10 times more sequences with genome locations as well as Gene Ontology annotations imported from QuickGO Release announcement

28 October 2018: NAR paper published
New RNAcentral paper has been published in Nucleic Acids Research.

5 January 2019: Release 11
RNAcentral now includes microRNA-target interactions from LncBase and TarBase as well as ncRNAs from Ensembl Plants Release announcement

11 June 2019: Release 12
RNAcentral now includes secondary structures for 3 million small subunit rRNAs and human lncRNAs from LncBook Release announcement

9 September 2019: Release 13
RNAcentral displays secondary structures for >7 million sequences, as well as adds the data from Ensembl Fungi, Protists, and Metazoa Release announcement

18 December 2019: Release 14
RNAcentral now contains LSU rRNA secondary structures and imports 5 new databases (5SrRNAdb, GeneCards, MalaCards, MirGeneDB, and snoDB) Release announcement

2 April 2020: Wellcome grant
RNAcentral and Rfam are funded by a grant from Wellcome.

21 May 2020: Release 15
RNAcentral has been updated with the latest data including links to interactions from the IntAct database Release announcement

18 September 2020: Release 16
RNAcentral now includes ZFIN, CRW, and snoRNA Database, as well as SO annotations and millions of new 2D structures Release announcement

27 October 2020: NAR paper published
New RNAcentral paper has been published in Nucleic Acids Research.

5 March 2021: Release 17
RNAcentral now includes piRBase as well as updated RNAse P diagrams Release announcement

9 June 2021: New paper is out in Nature Communications
A paper describing R2DT, a new method for RNA secondary structure visualisation, has been published in Nature Communications.

25 June 2021: Release 18
RNAcentral includes an updated rRNA classification and >22 million RNA secondary structures Release announcement

18 October 2021: Release 19
RNAcentral now includes PSICQUIC database as well as CPAT scores for lncRNAs and a major update to GtRNAdb Release announcement

31 March 2022: Release 20
RNAcentral has developed a text mining pipeline to automatically connect RNAcentral sequences and genes with the latest scientific literature Release announcement

17 October 2022: Release 21
RNAcentral now includes PlncDB and Expression Atlas as well as an update to LitScan Release announcement

15 February 2023: Release 22
RNAcentral now includes EVlncRNAs - a database of experimentally validated lncRNAs; Ribocentre - a database of ribozymes with representative structures; and now features a widget to visualise expression data from Expression Atlas Release announcement

1 November 2023: Release 23
RNAcentral now uses LLMs to generate summaries for selected ncRNA genes. This release also features a new genome browser (IGV.js) and two new expert databases, REDIportal and MGnify. Release announcement

11 March 2024: Release 24
RNAcentral release 24 features a significant update to tmRNA, expands text search capabilities to include lineage search, and incorporates over 100,000 new articles Release announcement

19 June 2025: Release 25
RNAcentral release 25 features a major update to Rfam, update TarBase to v9 and improvements to LitScan Release announcement

8 October 2025: Release 26
RNAcentral release 26 features the creation of **gene-level entries** for all sequences across 204 organisms Release announcement

29 July 2026: Release 27
RNAcentral release 27 features 5 new databases, including for the first time >5 million circRNAs Release announcement

16 October 2026: Release 28
RNAcentral release 28 contains over 58 million sequences from 59 expert databases Release announcement

2.2 Frequently Asked Questions

2.2.1 What is RNAcentral?

RNAcentral is an open public resource that offers integrated access to a comprehensive and up-to-date set of non-coding RNA (ncRNA) sequences. It assigns identifiers to distinct ncRNA sequences and automatically updates links between those sequences and the identifiers maintained by *expert databases*.

More about RNAcentral →

2.2.2 What are RNAcentral identifiers?

Each sequence in RNAcentral is assigned a **U**nique **RNA** **S**equence identifier (**URS**), for example `URS00000478B7`. These identifiers are stable and are not expected to change. **Species-specific identifiers** also include an NCBI taxid, such as URS00000478B7_9606.

More about identifiers →

2.2.3 How do I find an RNAcentral identifier?

For a **single sequence**, use RNAcentral *sequence search*. For a **large number of sequences**, use the *RNAcentral API* or the md5 and id mapping files in the FTP archive.

More about finding identifiers →

2.2.4 What sequences are excluded from RNAcentral?

- sequences shorter than **10 nucleotides**
- sequences with **more than 10% of unknown characters** (Ns).

2.2.5 How do I submit sequences to RNAcentral?

The easiest way is to submit your sequences to an INSDC database like ENA, GenBank, or DDBJ. They will then automatically appear in a subsequent RNAcentral release.

If you run an ncRNA database and would like to join the RNAcentral Consortium, please *get in touch*.

2.2.6 How often is RNAcentral updated?

We aim to update the RNAcentral data **every 3-6 months**, while the user interface and website functionality is continuously updated. See *Releases* for what changes between releases and how to get archived snapshots.

2.2.7 Why is my article not linked to an RNAcentral sequence?

LitScan links RNAcentral sequences to open access articles in Europe PMC. An article may be missing because:

- it was published after the last RNAcentral release, in which case it will be added in the next one;
- it doesn't mention any of the exact Ids, gene names or synonyms that LitScan searches for;
- it is not open access in Europe PMC, so LitScan can't read the full text.

2.2.8 Why are the citation counts different from Google Scholar, Web of Science or Scopus?

The citation counts shown for each article come from Europe PMC, which does not index the same content as Google Scholar, Web of Science or Scopus. Highly cited articles in Europe PMC are nonetheless generally highly cited on these platforms too.

2.2.9 What is the licence for the RNAcentral data?

The content on the RNAcentral website is licensed under a Creative Commons Zero licence, which means that you can use the data in any way and for any purpose.

More about the licence →

2.3 RNAcentral identifiers

Each sequence in RNAcentral is assigned a **U**nique **R**NA **S**equence identifier (**URS**). These identifiers are stable and are not expected to change.

2.3.1 URS identifiers

The identifiers have the following format: `URS + sequentially assigned hexadecimal number` and can be parsed using this regular expression:

```
/URS[0-9A-F]{10}/
```

Example identifiers:

- `URS0000000001`
- `URS00000478B7`

2.3.2 Species-specific identifiers

The same RNA sequence can occur in more than one species. **Species-specific identifiers** also include an NCBI taxid, for example 9606 for *Homo sapiens*.

Either a slash `/` or an underscore _ can be used as a separator:

- URS00000478B7_9606
- URS00000478B7/9606

Links without NCBI taxonomy identifiers show annotations from all species where the sequence was found. See *Linking to RNAcentral* for the full URL patterns.

2.3.3 Gene identifiers

RNAcentral also groups sequences into *genes*, each with its own identifier. Gene identifiers start with `RNAC`, followed by the species' Ensembl prefix and `G`, then a number and a version, for example RNACG65104808090.4 for the human *RN7SL1* gene. See *Gene identifiers* for how they are assigned between releases.

2.3.4 Stability

RNAcentral identifiers are intended to be stable. Links between RNAcentral sequences and identifiers maintained by *expert databases* are updated automatically as RNAcentral imports new releases from those databases.

2.3.5 Finding an identifier

To find an RNAcentral identifier for a **single sequence**, use RNAcentral *sequence search*.

For a **large number of sequences**, one can:

- use the *example script* on the REST API page, which looks up sequences by their MD5 hash;
- download a mapping file from the RNAcentral FTP site with correspondences between md5 values and RNAcentral ids;
- download a mapping file with correspondences between external database identifiers and RNAcentral ids.

You can also use the `/link/<database>:<accession>` URL pattern described in *Linking to RNAcentral*.

2.3.6 Excluded sequences

RNAcentral excludes:

- sequences shorter than **10 nucleotides**;
- sequences with **more than 10% of unknown characters** (Ns).

2.4 Expert Databases

Expert Databases are specialist resources that curate RNA data within their own area of expertise, such as a group of organisms or a type of RNA. RNAcentral imports their data, combines it into a single view of non-coding RNA sequences, and links each sequence back to the original records. The Expert Databases remain the authoritative source for their own curation.

2.4.1 RNAcentral Consortium

Together, the Expert Databases form the **RNAcentral Consortium**. Currently the consortium is formed by **66** Expert Databases, **58** of which have already been imported into RNAcentral. RNAcentral's *papers* are published under the name "The RNAcentral Consortium".

The members range from general sequence archives such as ENA and PDBe, through genome and model organism databases, to resources for a single type of RNA and for second-order data such as interactions and modifications.



2.4.2 What Expert Databases contribute

Depending on their scope, Expert Databases contribute sequences, gene names and descriptions, RNA types, functional annotations, RNA interactions and RNA modifications. Each record is linked to the *URS identifier* of its sequence, and sequence pages link back to the original records.

Each database has a landing page in RNAcentral showing what it contributes, for example FlyBase. To find all the sequences from one database, search for it by name, for example `expert_db:"FlyBase"`.

2.4.3 Members

The **Imported** column shows the databases whose data are already in RNAcentral.

Database	Imported	Description
5SrRNAdb	✓	5SrRNAdb is an information resource for 5S ribosomal RNAs
CircAtlas	✓	CircAtlas is a database of circular RNAs (circRNAs) that integrates expression, con
CIRCpedia	✓	CIRCpedia is a comprehensive circular RNA database that integrates large-scale sec

Table 1 – continued from

Database	Imported	Description
CRS		CRS is a database of conserved RNA motifs identified computationally in multi-spe
CRW	✓	CRW provides comparative sequence and structure information for ribosomal, intro
dictyBase	✓	dictyBase is the model organism database for the social amoeba Dictyostelium disco
ENA	✓	ENA is a comprehensive record of the world's nucleotide sequencing information
Ensembl	✓	Ensembl is a genome browser for vertebrate genomes and model organisms that sup
Ensembl Fungi	✓	Ensembl Fungi is a genome browser for fungi genomes that complements the Ensem
Ensembl Metazoa	✓	Ensembl Metazoa is a genome browser for metazoan genomes that complements the
Ensembl Plants	✓	Ensembl Plants is a genome browser for plant genomes that complements the Ensem
Ensembl Protists	✓	Ensembl Protists is a genome browser for protist genomes that complements the Ens
Ensembl/GENCODE	✓	GENCODE produces high quality reference gene annotation and experimental valid
EVLncRNAs	✓	A database of lncRNAs
Expression Atlas	✓	A database of expression
FlyBase	✓	FlyBase is a database of Drosophila genes and genomes
GeneCards	✓	GeneCards is a searchable, integrative database that provides comprehensive, user-f
Greengenes	✓	Greengenes is a database of full-length 16S rRNA gene that provides a curated taxo
GtRNAdb	✓	GtRNAdb contains tRNA gene predictions on complete or nearly complete genome
HGNC	✓	HGNC is the worldwide authority that assigns standardised nomenclature to human
IntAct	✓	IntAct provides a freely available, open source database system and analysis tools fo
JaponicusDB	✓	JaponicusDB is the model organism database for Schizosaccharomyces japonicus, in
LncBase	✓	LncBase provides experimentally verified and computationally predicted microRNA
LncBook	✓	LncBook is a curated knowledgebase of human long non-coding RNAs
LNCipedia	✓	LNCipedia is a comprehensive compendium of human long non-coding RNAs
lncRNAdb	✓	lncRNAdb is a database providing comprehensive annotations of eukaryotic long no
MalaCards	✓	MalaCards integrates manually-curated and text-mining sources to associate genes, i
MGI	✓	MGI is the international database resource for the laboratory mouse
MGnify	✓	MGnify is a hub for the analysis and exploration of metagenomic, metatranscriptom
miRBase	✓	miRBase contains high-quality miRNA annotations; miRBase is responsible for assi
MirGeneDB	✓	MirGeneDB is a curated microRNA gene database covering 45 metazoan organisms
miRTarBase		miRTarBase is an experimentally validated microRNA-target interactions database
MirtronDB	✓	MirtronDB is a database of mirtrons, small intronic microRNA precursors identified
Modomics	✓	Modomics is a comprehensive database of RNA modifications
NONCODE	✓	NONCODE is an integrated knowledge database dedicated to non-coding RNAs
NPInter		NPInter contains data on experimentally determined functional interactions between
PDBe	✓	PDBe is the European repository of information about the 3D structures of large bio
piRBase	✓	piRBase is a database of various piRNA associated data to support piRNA function
PLncDB	✓	PLncDB provides comprehensive genomic view of plant lncRNAs
PomBase	✓	PomBase is a comprehensive database for the fission yeast Schizosaccharomyces pom
PSICQUIC	✓	A database of manually annotated human miRNA interactions
RDP	✓	RDP provides quality-controlled, aligned and annotated rRNA sequences and a suite
REDIportal	✓	An ATLAS of A-to-I RNA editing events in human and other organisms
RefSeq	✓	RefSeq is a comprehensive, integrated, non-redundant, well-annotated set of referen
Rfam	✓	Rfam is a collection of non-coding RNA families, represented by manually curated
RGD	✓	RGD is a rat genetic and genomic research resource
Ribocentre	✓	A database of ribozymes with representative structures
RiboVision	✓	A database of ribosomal annotations
RNApathwaysDB		RNApathwaysDB contains RNA maturation and decay pathways
Sci-ModoM	✓	Sci-ModoM is a quantitative database of transcriptome-wide high-throughput RNA
SGD	✓	SGD provides comprehensive integrated biological information for the budding yeas

Table 1 – continued from

Database	Imported	Description
SILVA	✓	SILVA is a comprehensive resource for quality checked and aligned ribosomal RNA
snoDB	✓	snoDB is an interactive database of human snoRNA sequences, abundance and inter
snOPY	✓	snOPY provides comprehensive information about snoRNAs, their gene loci, orthol
snoRNA Atlas		snoRNA Atlas is a database of human snoRNAs
snoRNA Database	✓	The snoRNA Database is a curated collection of archaeal snoRNAs maintained by t
sRNAmap		sRNAmap is a collection of sRNA sequences and interactions
SRPDB	✓	SRPDB provides aligned, annotated and phylogenetically ordered sequences related
TAIR	✓	TAIR is a database of genetic and molecular biology data for the model higher plant
TarBase	✓	TarBase is a collection of manually curated experimentally validated miRNA-gene i
tmRDB		tmRDB is a collection of aligned, annotated and phylogenetically ordered sequences
tmRNA Website	✓	tmRNA Website contains predicted tmRNA sequences from RefSeq bacterial genor
tRNAdb		tRNAdb is a compilation of tRNA sequences and tRNA genes.
WormBase	✓	WormBase curates, stores and displays genomic and genetic data about nematodes v
ZFIN	✓	The Zebrafish Information Network (ZFIN) is the database of genetic and genomic
ZWD	✓	ZWD is a git-based collection of non-coding RNA alignments maintained by Dr Za

2.4.4 Joining the consortium

If you develop a non-coding RNA database and would like to join the RNAcentral Consortium, please *contact us*. To add individual sequences instead, see *How do I submit sequences to RNAcentral?*

2.5 Releases

We aim to update the RNAcentral data **every 3-6 months**, while the user interface and website functionality are continuously updated.

Release announcements are published on the RNAcentral blog. See the *Timeline* for a release-by-release history of the project.

2.5.1 Current and archived releases

The *FTP Archive* provides release snapshots:

- `current_release/` contains the latest RNAcentral release files;
- `releases/` contains previous release snapshots, back to version 1.0-beta (2014).

Use archived release files when you need reproducible analyses based on a specific RNAcentral data release.

2.5.2 What changes between releases

RNAcentral releases can include changes such as:

- newly imported or updated *expert database* records;
- new *RNAcentral identifiers* for newly observed sequences;
- updated links between RNAcentral sequences and expert database accessions;
- updated genome coordinate files;
- recomputed *genes*, whose identifiers may change version or be retired;
- new *secondary structure* diagrams made with R2DT;

- updated functional annotations, including *Rfam quality control* and *Gene Ontology* annotations;
- updated sequence FASTA files and database dumps;
- new *LitScan* publications and *LitSumm* summaries.

2.6 Publications

2.6.1 How to cite

Attribution isn't required, but **we recommend citing RNAcentral** when you use its services (such as sequence search), data or software. It helps us show how RNAcentral is used and secure funding for its ongoing development and maintenance.

Recommended citation

RNAcentral in 2026: genes and literature integration
The RNAcentral Consortium, 2025. *Nucleic Acids Research (Database issue)*.
NAR | EuropePMC | PubMed

If your work relies on a specific RNAcentral tool, feature or data release, cite the paper below that best describes it as well.

See *Licence* for the terms the data itself is released under.

2.6.2 Previous RNAcentral publications

RNAcentral has also been described in these earlier papers:

- **RNAcentral 2021: secondary structure integration, improved sequence search and new member databases**
  The RNAcentral Consortium, 2020. *Nucleic Acids Research (Database issue)*.
  NAR | EuropePMC | PubMed
- **Exploring Non-Coding RNAs in RNAcentral**
  Sweeney et al., 2020. *Current Protocols in Bioinformatics*.
  CPB | EuropePMC | PubMed
- **RNAcentral: a hub of information for non-coding RNA sequences**
  The RNAcentral Consortium, 2019. *Nucleic Acids Research (Database issue)*.
  NAR | EuropePMC | PubMed
- **RNAcentral: a comprehensive database of non-coding RNA sequences**
  The RNAcentral Consortium, 2017. *Nucleic Acids Research (Database issue)*.
  NAR | EuropePMC | PubMed
- **RNAcentral: an international database of ncRNA sequences**
  The RNAcentral Consortium, 2015. *Nucleic Acids Research (Database issue)*.
  NAR | EuropePMC | PubMed
- **RNAcentral: A vision for an international database of RNA sequences**
  Bateman et al., 2011. *RNA*.
  RNA | EuropePMC | PubMed

2.6.3 Software

Papers describing software developed by the RNAcentral team:

- **GOFlowLLM: curating miRNA literature with large language models and flowcharts**
  Green et al., 2026. *Bioinformatics*.
  Bioinformatics | EuropePMC | PubMed
- **LitSumm: large language models for literature summarization of noncoding RNAs**
  Green et al., 2025. *Database*.
  Database | EuropePMC | PubMed
- **R2DT: a comprehensive platform for visualizing RNA secondary structure**
  McCann et al., 2025. *Nucleic Acids Research*.
  NAR | EuropePMC | PubMed
- **R2DT is a framework for predicting and visualising RNA secondary structure using templates**
  Sweeney et al., 2021. *Nature Communications*.
  Nature Communications | EuropePMC | PubMed

2.7 Training resources

2.7.1 Online training course

If you would like to find out more about RNAcentral or take a tour of the website, consider taking an online training course that contains a data overview, tips for searching, quizzes, and more.

Take the course now →

2.7.2 Webinars

Accessing RNA data programmatically with the Rfam and RNAcentral APIs

Recorded on 25 June 2026.

Using RNAcentral to explore and investigate non-coding RNA sequences

Watch a recording of an online webinar to see RNAcentral in action:

Recorded on 21 September 2023.

2.7.3 Other resources

Explore community-contributed educational resources including tutorials, slides, and lecture materials: Educational Resources Repository.

2.8 Scientific Advisory Board

2.8.1 Current members

RNAcentral and Rfam share a Scientific Advisory Board (SAB) of RNA biologists covering a wide range of expertise from wet lab to computational research.

- Alain Laederach, UNC Chapel Hill, USA
- Madeline Sherlock, UC Santa Cruz, USA

- Peter Stadler, Universität Leipzig, Germany
- Irmtraud Meyer, Max Delbrück Centre, Germany
- Chun Kit Kwok, City University of Hong Kong, Hong Kong

The SAB oversees the progress of the project during annual meetings.

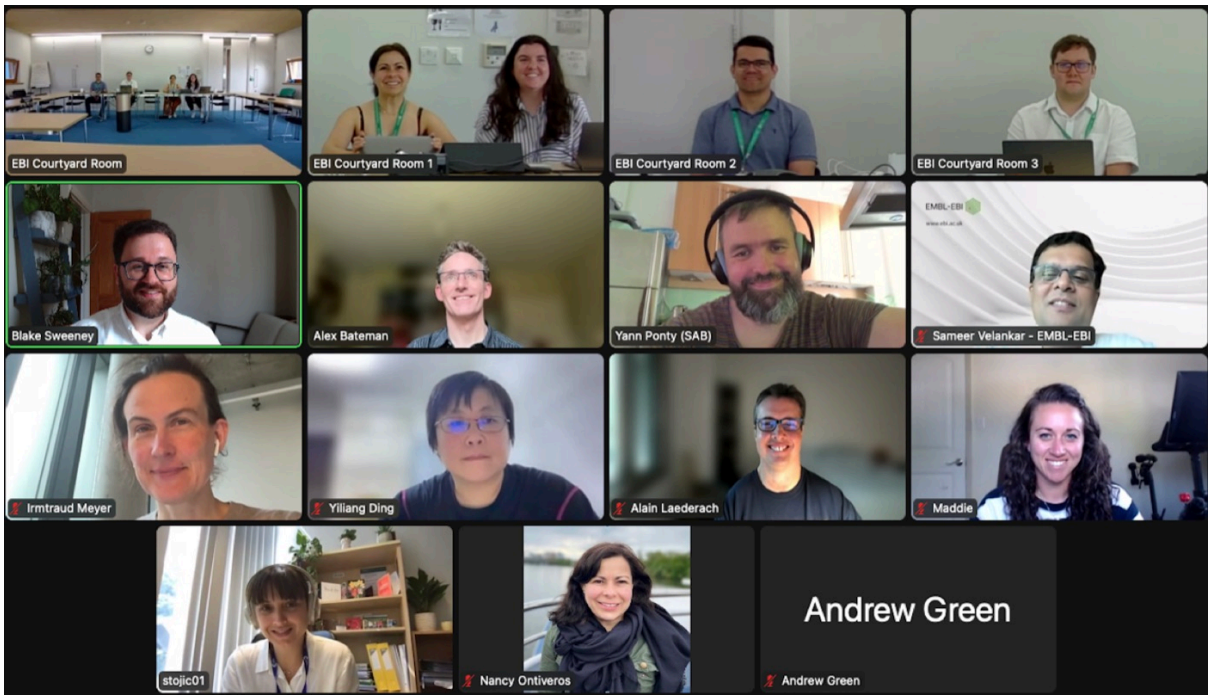


Fig. 1: SAB meeting in 2023

2.8.2 Previous members

- Sean Eddy (Harvard University), *SAB member between 2013-2017*
- Mihaela Zavolan (University of Basel), *SAB member between 2016-2020*
- Eric Westhof (University of Strasbourg), *SAB member between 2015-2021*
- Michelle Meyer (Boston College), *SAB member between 2016-2021*
- Manja Marz (Friedrich Schiller University Jena, Germany), *SAB member between 2017-2022*
- Yiliang Ding (John Innes Centre, UK), *SAB member between 2021-2023*
- Yann Ponty (CNRS École Polytechnique, France), *SAB member between 2019-2024*
- Lovorka Stojic (Barts Centre CRUK, UK), *SAB member between 2020-2025*

2.9 Meet the Team

2.9.1 Current members

The RNAcentral database is developed and maintained at the European Bioinformatics Institute in Cambridge, UK.

- Pedro Madrigal, Project Leader

- Alex Bateman, Group Team Leader
- Philippa Muston, Full Stack Developer
- Isaac Jandalala, Full Stack Developer
- Andrew Green, ARISE fellow
- Victoria Begley, Bioinformatician
- Hanh Hoang, Bioinformatics Data Engineer

2.9.2 Previous members

- Anton Petrov, Project Leader
- Blake Sweeney, Project Leader
- Boris Burkov, Software Developer
- Carlos Ribas, Full Stack Developer

2.10 Text search

The search box on every RNAcentral page searches sequence descriptions, genes, species, expert databases and publications. Plain keywords are often enough; the syntax below narrows the results.

2.10.1 Exact matching

Use double quotes (**“”**) to search for exact matches.

Example: `"hsa-mir-21"` will find only *hsa-mir-21* and not *hsa-mir-212*.

2.10.2 Wildcards

A wildcard character (*) can match any number of characters. Wildcards are added automatically to search terms that are **not** enclosed in double quotes, except terms shorter than three characters, DOIs and Rfam family identifiers.

Example: a search for `HOTAIR` (no double quotes) will find both *HOTAIR* and *HOTAIRM1* genes, and a search for `"HOTAIR"` (with double quotes) will find only *HOTAIR*.

2.10.3 Field-specific search

Search can be restricted to specific fields using the **field_name:”field value”** syntax. Please note that “field value” **must be enclosed in double quotes**.

Field	Examples
expert database	`expert_db:"tmrna website"`, `expert_db:"mirbase"` (the "Expert databases" facet on any results page lists the names)
NCBI taxonomic identifier	`taxonomy:"9606"` where 9606 is the NCBI taxonomy id for *Homo sapiens*
taxonomy string	`tax_string:"primates"` (searches a whole taxonomic group)
scientific species name	`species:"Mus musculus"` or `tax_string:"Mus musculus"`
common species name	`tax_string:"mouse"`
RNA type	`rna_type:"pirna"` or `so_rna_type_name:"pirna"` (the latter is classified using Sequence Ontology)
gene	`gene:"hotair"`
organelle	`organelle:"mitochondrion"`, `organelle:"plastid"`
description	`description:"16S"`
length	`length:"1500"`, `length:[9000 TO 10000]` (supports range queries)
publication title	`pub_title:"Danish population"`
author	`author:"Girard A."`
PubMed id	`pubmed:"17881443"`
Digital Object Identifier	`doi:"10.1093/nar/19.22.6328"`
MD5	`md5:"020711a90d35bb197e29e085595dd52e"` (the MD5 hash of the sequence written as uppercase DNA)
interacting protein	`interacting_protein:"ENSG00000277791"`
interacting RNA	`interacting_rna:"ENSG00000235652"`
evidence for interaction	`evidence_for_interaction:"ago-ip"`
secondary structure	`has_secondary_structure:"True"`
conserved structure	`has_conserved_structure:"True"`
GO annotation	`has_go_annotations:"True"`
genome location	`has_genomic_coordinates:"True"`
any interacting protein	`has_interacting_proteins:"True"`
any interacting RNA	`has_interacting_rnas:"True"`
LitScan publications	`has_lit_scan:"True"`
LitSumm AI summary	`has_litsumm:"True"`
RNA editing events	`has_editing_event:"True"`

2.10.4 Logic operators

Operators can be typed in upper or lower case.

- **AND** (default)

  Multiple search terms separated by white spaces are combined using AND, so a query like `Homo sapiens` is treated as `Homo AND sapiens` and only entries having both terms will be found.

- **OR** (to indicate equivalence)

  Example: `rna_type:"pirna" OR rna_type:"mirna"`

- **NOT** (to indicate exclusion)

  Example: `expert_db:"lncrnadb" NOT expert_db:"rfam"`

2.10.5 Grouping

Use parentheses to group and nest logical terms.

Example: `(expert_db:"mirbase" OR expert_db:"lncrnadb") NOT expert_db:"rfam"`

2.10.6 Search tips

- Make sure your **spelling** is correct.

  Example: misspelled terms like `Esherichia` (missing "c") won't find any results.

- Use **full species names**.

  Example: use `Escherichia coli` and not `E. coli` as your search terms.

2.10.7 Download search results

The RNAcentral text search supports exporting any number of search results in the following formats:

- FASTA
- Parquet
- JSON
- RNAcentral IDs
- Hugging Face Dataset (requires a Hugging Face account)

Also refer to the *public Postgres database* and the *RNAcentral FTP Archive* for exporting large amounts of data.

2.10.8 Step-by-step walkthrough

Our Current Protocols in Bioinformatics article describes different ways of accessing the data, including a walkthrough of the text search.

2.10.9 Searching programmatically

RNAcentral search is powered by EBI Search, which provides a public REST interface to the same index under the `rnacentral` domain.

The website adjusts each query before sending it to EBI Search, so queries sent to the REST interface directly need a few changes:

- write `AND`, `OR`, `NOT` and the `TO` in length ranges in upper case;
- write `taxonomy`, `pubmed` and `doi` in upper case, as `TAXONOMY:"9606"`;
- use `tax_string` in place of `species`;
- add wildcards yourself, as `HOTAIR*`.

Example: human piRNAs as JSON, using the query `rna_type:"pirna" AND TAXONOMY:"9606"`.

Related pages:

- *Sequence search*
- *REST API*

2.11 Sequence search

The RNAcentral sequence similarity search enables searches against a comprehensive collection of non-coding RNA sequences from a consortium of *RNA databases*.

2.11.1 Powered by nhmmer

The search is powered by the nhmmer software which is more sensitive than *blastn* but is comparable in speed. The command executed by the sequence search is as follows:

```
nhmmer --qfasta --tformat fasta -o <output_file> -T 0 --F3 0.02 --rna --
↪watson --cpu 4 -Z <number> <query> <database.fasta>
```

Where:

- **–F3 0.02** is only used when the sequence length is less than 50 nucleotides.
- **-Z <number>** is calculated according to the size of the database.

Each parameter is described in the HMMER User Guide.

2.11.2 Query length

Queries must be between **10 and 7,000 nucleotides** long.

2.11.3 Exact sequence matches

Whenever a sequence is entered in the search input box, the query is compared with all RNAcentral sequences and if there is an **exact match**, the links to the entries matching the query are displayed in a green "Identical match" box. This is very quick because only identical matches are considered. To see all similar sequences, just click Submit.

2.11.4 Searching for ribosomal RNAs

Over 50% of RNAcentral sequences are ribosomal RNAs (rRNAs). The abundance and high conservation of rRNA sequences makes it difficult to perform sequence similarity searches, as such searches are expected to match a large number of sequences and can take a long time to complete.

To get around this, the sequence similarity searches are performed against a subset of ~100,000 rRNA sequences from Ensembl, FlyBase, HGNC, MGI, PDBe, PomBase, RDP, RefSeq, RGD, SGD, TAIR, and WormBase.

2.11.5 Batch queries

The web interface accepts up to **50 sequences** per submission, so use the *API* for larger batches.

2.11.6 Rfam classification

In addition to nhmmer searches against RNAcentral, every query is automatically compared with the Rfam library of RNA families. The searches are done using the Infernal cmscan program coupled with a post-processing step that removes hits overlapping Rfam families from the same clan (a clan is a set of homologous families, for example LSU_rRNA_archaea, LSU_rRNA_bacteria and LSU_rRNA_eukarya). The Rfam website and the EBI cmscan service report these redundant hits.

2.11.7 Secondary structure

Every search also sends your query sequence to *R2DT*, which selects the best-matching template from its library of ~4,000 RNA structures, folds the query into that template's secondary structure (2D) and draws it in the template's standard layout, so the query can be compared with known structures of the same type. The diagram appears in the **Secondary structure** section of the results page, and **View** opens it in the R2DT viewer, where it can be explored and downloaded. If the query does not match any R2DT template, no diagram is shown.

2.11.8 Number of similar sequences

Although the number of similar sequences can reach tens of thousands, for performance reasons, only the **top 1000 results** will be shown in each search.

2.11.9 Download search results

The search results can be downloaded as a ZIP file containing:

- *datapackage.json*, with the search metadata;
- *sequences/similar-sequences.json* and *sequences/similar-sequences.txt*, with the description, E-value, identity, query coverage, gaps and alignment of each match (the text file also gives each match's RNAcentral ID);
- *sequences/similar-sequences.fasta*, with the aligned region of each matching sequence, named as `URS…/start-end`.

2.11.10 How long are the search results available for?

The results will be available at the same URL for **up to 7 days**. After this period, **all data and metadata are deleted**.

2.12 Sequence search API

The RNAcentral sequence similarity search API allows you to perform most of the same actions through your own tools as you can through the *web interface*. Any tool capable of making HTTP requests can communicate with this API, for example `curl`.

Base URL: https://sequence-search.rnacentral.org/

The web interface accepts up to 50 sequences per submission, so use the API for larger batches.

2.12.1 How it works

The API is **asynchronous**:

1. **Submit** a job with `POST /api/submit-job`, passing the `sequence` (10 to 7,000 nucleotides) and an optional list of *expert databases* to search, such as `["mirbase"]`. Leave the list empty to search all of RNAcentral. The response contains a `job_id`.
2. **Poll** `GET /api/job-status/<job_id>` until the `status` is `finished` (or `error` or `not_found`). The response also includes a `progress` percentage.
3. **Retrieve** results from `GET /api/job-results/<job_id>`, which is paginated with `page` and `page_size` parameters (100 per page by default) and reports the total `hit_count`. Each entry in `entries` includes the `rnacentral_id`, `description`, `e_value`, `identity`, `query_coverage` and `alignment`. Unlike the web interface, which shows the top 1,000 hits, the API returns them all.

2.12.2 API throttling

Job submissions (`/api/submit-job`) are rate limited to **10 requests per minute per IP address**. There is **no rate limit** on job status or results endpoints.

When the service is busy, a submission can also be refused with HTTP status `429` and a message asking you to wait. Wait a minute and submit again; the example script below does this automatically.

2.12.3 Example script

An example of a Python script to perform searches for each record of a FASTA file is provided below. It requires Python 3.5 or later and the `biopython` and `requests` packages (`pip install biopython requests`).

You can use this script to search for sequences in the RNAcentral database or in one of the *Expert Databases*. The results will be organized in a new directory. A file in JSON format containing the metadata and the search results will be created. This file will be named with the description of the sequence. Here is an example:

```
results
|-- URS00001F7A66_9031 Gallus gallus (chicken) gga-let-7g-5p.json
|-- ...
```

```
"""
Usage: python search.py [file] [database]

Examples:
python search.py file.fasta                # search in all databases
python search.py file.fasta mirbase        # search in miRBase only
"""
import json
import requests
import sys
import time
from Bio import SeqIO
from pathlib import Path
```

(continues on next page)

(continued from previous page)

```
SERVER = "https://sequence-search.rnacentral.org/"


def get_sequence_search_result(description, sequence, job_id):
    """Poll job status then retrieve all paginated sequence search results."
↪""
    job_status = "pending"
    while True:
        try:
            data = requests.get(SERVER + "api/job-status/" + job_id,␣
↪timeout=30).json()
            job_status = data["status"]
            progress = data.get("progress", 0)
        except requests.exceptions.RequestException as e:
            print("Error checking job status, retrying: {}".format(e))
            time.sleep(10)
            continue

        if job_status == "finished":
            break
        elif job_status in ("error", "not_found"):
            print("Search failed for record: {}".format(description))
            return {
                "job_id": job_id,
                "status": job_status,
                "description": description,
                "sequence": str(sequence),
            }
        else:
            print("  {:.0f}% complete...".format(progress))
            time.sleep(10)

    # Fetch results page by page
    results = []
    page = 1
    page_size = 100
    hit_count = None
    while True:
        try:
            url = "{}api/job-results/{}?page={}&page_size={}".format(
                SERVER, job_id, page, page_size
            )
            data = requests.get(url, timeout=30).json()
        except requests.exceptions.RequestException as e:
            print("Error fetching results page {} for {}: {}".format(page,␣
↪description, e))
            break

        if hit_count is None:
```

(continues on next page)

(continued from previous page)

```
            hit_count = data.get("hit_count", 0)
        entries = data.get("entries", [])
        results.extend(entries)
        if not entries or len(results) >= hit_count:
            break
        page += 1

    return {
        "job_id": job_id,
        "hits": hit_count or 0,
        "status": job_status,
        "description": description,
        "sequence": str(sequence),
        "results": results,
    }


def main():
    if len(sys.argv) == 1:
        print("You must specify the FASTA file")
        exit()
    elif len(sys.argv) == 2:
        filename = sys.argv[1]
        database = []
    elif len(sys.argv) == 3:
        filename = sys.argv[1]
        database = [sys.argv[2]]
    else:
        print("Usage: python search.py file.fasta [database]")
        exit()

    Path("results").mkdir(parents=True, exist_ok=True)

    with open(filename, mode="r") as handle:
        for record in SeqIO.parse(handle, "fasta"):
            description = record.description
            sequence = record.seq
            data = {"databases": database, "sequence": str(sequence)}
            post_job = requests.post(SERVER + "api/submit-job", json=data)
            while post_job.status_code == 429:
                print("Rate limited, waiting a minute before resubmitting...
↪")
                time.sleep(60)
                post_job = requests.post(SERVER + "api/submit-job",␣
↪json=data)

            job_id = None
            if post_job.status_code == 200:
                job_id = post_job.json()["job_id"]
```

(continues on next page)

(continued from previous page)

```
            else:
                print("Failed to submit job for record:\n{}\n{}".
↪format(description, sequence))

            if job_id:
                result = get_sequence_search_result(description, sequence,␣
↪job_id)
                if result:
                    hits = result.get("hits", 0)
                    if hits == 0:
                        print("No results found for {}".format(description))
                    else:
                        print("Saving {} hit(s) for {}".format(hits,␣
↪description))
                        with open("results/" + description + ".json", "w")␣
↪as f:
                            json.dump(result, f)


if __name__ == "__main__":
    main()
```

2.12.4 Results retention

Results are kept for **up to 7 days**, after which all data and metadata are deleted. They can be removed sooner when the service is under heavy load, so save the results once a job finishes.

2.13 REST API

Most data in RNAcentral can be accessed programmatically using a RESTful API allowing for integration with other resources. The API implementation is based on the Django REST framework.

For an introduction, watch the *webinar on the Rfam and RNAcentral APIs*.

2.13.1 Web browsable API

The RNAcentral API is **web browsable**, which means that:

- the data is served in either **human- or computer-friendly formats** depending on whether a URL is viewed in a browser or is retrieved programmatically;
- many resources are **hyperlinked** so that it's possible to navigate the API in the browser.

As a result, developers can familiarise themselves with the API and get a better sense of the RNAcentral data.

Browse the API →

2.13.2 Versioning

To ensure that changes in the RNAcentral API don't break the applications relying on it, the API is versioned, and **the version is included in the API's URL**.

For example, the **current API is at Version 1** and is available at https://rnacentral.org/api/v1, and the next version will be available at `https://rnacentral.org/api/v2`.

The **latest version** of the API can be accessed at https://rnacentral.org/api/current, but it's not recommended to use this endpoint for long-term development as the underlying data model may change.

No **backward-incompatible changes** are made to each version after it's been made public. More specifically, it's guaranteed that within one version there will be no:

- changing urls
- deleting or renaming data fields
- changing data field types

The following **non-disruptive changes** may be implemented to a public API:

- adding new endpoints
- adding new data fields
- adding new filtering methods

An advance notice will be given before obsoleting an API version. To stay up to date, please consider signing up for the RNAcentral updates.

2.13.3 Throttling

The maximum number of requests from the same IP address is limited to **5 requests per second** for most endpoints, and **30 requests per minute** for endpoints under `/api/v1/rna/`. Currently, there is no limit on the total number of requests from the same IP.

Please *get in touch* if you require a higher request rate.

2.13.4 Responsible automated access

RNAcentral is a free public resource. This guidance applies to scripts, crawlers, and AI agents, as well as other automated clients. It helps keep the service available to everyone.

Choose the appropriate access method

- Use this REST API for targeted, programmatic lookups.
- For whole datasets, release-scale analyses, or large ID mappings, first check the *FTP bulk downloads*, which lists the available files and formats, or the RNAcentral Hugging Face datasets. Do not reconstruct a release or download a large dataset by fetching individual API records.
- AI assistants that support the *RNAcentral MCP server* should use it for natural-language research tasks instead of scraping HTML pages or coordinating many raw API calls.
- Use the *sequence-search API* only for necessary similarity searches; it has separate submission limits.

Suggested request behaviour

These are suggested operating rates, not additional quota allocations. Do not exceed the enforced limits above.

Service	Suggested behaviour	Enforced limit
Most REST API endpoints	Keep sustained traffic at or below **1 request per second** from one client/IP.	5 requests per second per IP
`/api/v1/rna/` endpoints	Keep traffic below the limit; a conservative client makes no more than **one request every 2 seconds**.	30 requests per minute per IP
Sequence-search job submissions	Submit only required searches and do not retry immediately when the service is busy.	*10 requests per minute per IP*

In addition:

- Cache responses where appropriate, reuse results, and narrow queries with filters before requesting more pages.
- Request no more than 100 entries per page and fetch pages sequentially. Avoid `flat=true` unless nested data are needed, because those responses can take longer to generate.
- On HTTP `429`, honour `Retry-After` when it is present. Otherwise, use exponential backoff with jitter; do not retry concurrently or rotate IP addresses to bypass a limit.
- Identify automated requests with a descriptive `User-Agent`, ideally including a contact URL or email address. Do not impersonate a browser or another client.
- Do not scrape the website when a documented API, bulk download, or MCP tool provides the data. Follow applicable `robots.txt` rules when crawling public pages.

If your work needs more capacity, a different access pattern, or long-running automated access, please *contact us* before starting it.

2.13.5 API v1 documentation

The sections below document **version 1** of the API, currently the only public version.

Example responses

Responses containing **multiple entries** have the following fields:

- *next* and *previous* are urls to the corresponding results page
- *results* is an array of entries.

Example: https://rnacentral.org/api/v1/rna/?md5=06808191a979cc0b933265d9a9c213fd

```
{
    "next": "https://rnacentral.org/api/v1/rna/?
↪md5=06808191a979cc0b933265d9a9c213fd&page=2",
    "previous": null,
    "results": [
        {
            "url": "https://rnacentral.org/api/v1/rna/URS00002C9E9D",
            "rnacentral_id": "URS00002C9E9D",
```

(continues on next page)

(continued from previous page)


```
            "md5": "06808191a979cc0b933265d9a9c213fd",
            "sequence":
↪"CUGAAUAAAUAAGGUAUCUUAUAUCUUUUAAUUAACAGUUAAACGCUUCCAUAAAGCUUUUAUCCA",
            "length": 66,
            "xrefs": "https://rnacentral.org/api/v1/rna/URS00002C9E9D/xrefs
↪",
            "publications": "https://rnacentral.org/api/v1/rna/
↪URS00002C9E9D/publications",
            "is_active": true,
            "description": "tRNA from 1 species",
            "rna_type": "tRNA",
            "count_distinct_organisms": 1,
            "distinct_databases": ["ENA"],
            "distinct_organisms": [7769]
        }
    ]
}
```


Responses containing just a **single entry** don't have the extra navigation fields:

Example: https://rnacentral.org/api/v1/rna/URS00002C9E9D


```
{
    "url": "https://rnacentral.org/api/v1/rna/URS00002C9E9D",
    "rnacentral_id": "URS00002C9E9D",
    "md5": "06808191a979cc0b933265d9a9c213fd",
    "sequence":
↪"CUGAAUAAAUAAGGUAUCUUAUAUCUUUUAAUUAACAGUUAAACGCUUCCAUAAAGCUUUUAUCCA",
    "length": 66,
    "xrefs": "https://rnacentral.org/api/v1/rna/URS00002C9E9D/xrefs",
    "publications": "https://rnacentral.org/api/v1/rna/URS00002C9E9D/
↪publications",
    "is_active": true,
    "description": "tRNA from 1 species",
    "rna_type": "tRNA",
    "count_distinct_organisms": 1,
    "distinct_databases": ["ENA"],
    "distinct_organisms": [7769]
}
```


Hyperlinked vs flat responses

Some objects are represented by hyperlinks, for example in the default RNA object cross-references are represented by a hyperlink:


```
"xrefs": "https://rnacentral.org/api/v1/rna/URS0000000001/xrefs/"
```


Sometimes it is desirable to retrieve all nested objects in a single request, which can be done using the `flat=true` url parameter. Note that such requests may take longer.

Examples

- https://rnacentral.org/api/v1/rna/ (hyperlinked)
- https://rnacentral.org/api/v1/rna/?flat=true (flat)
- https://rnacentral.org/api/v1/rna/URS0000000001 (hyperlinked)
- https://rnacentral.org/api/v1/rna/URS0000000001/?flat=true (flat)

Pagination

Responses containing multiple entries are paginated to prevent accidental downloads of large amounts of data and to speed up the API.

To navigate between pages, specify the `page` parameter.

The page size is controlled by the `page_size` parameter. Its default value is **10 entries per page**. Use pages of up to **100 entries**: larger pages are much slower and can time out.

Examples

- https://rnacentral.org/api/v1/rna/?page_size=5
- https://rnacentral.org/api/v1/rna/?page=2&page_size=5

Output formats

The following output formats are supported for all endpoints: **JSON**, **YAML** and **HTML** (the browsable API).

In addition, the data for individual RNA sequences can be downloaded in **FASTA** format.

There are three ways of specifying the format:

1. `format` url parameter
    - https://rnacentral.org/api/v1/rna/URS0000000001/?format=json
    - https://rnacentral.org/api/v1/rna/URS0000000001/?format=fasta
    - https://rnacentral.org/api/v1/rna/URS0000000001/?format=yaml
    - https://rnacentral.org/api/v1/rna/URS0000000001/?format=api
2. `.format` suffix
    - https://rnacentral.org/api/v1/rna/URS0000000001.json
    - https://rnacentral.org/api/v1/rna/URS0000000001.fasta
    - https://rnacentral.org/api/v1/rna/URS0000000001.yaml
    - https://rnacentral.org/api/v1/rna/URS0000000001.api
3. Accept headers

```
curl -H "Accept: application/json" https://rnacentral.org/api/v1/rna/
↪URS0000000001
curl -H "Accept: text/fasta" https://rnacentral.org/api/v1/rna/
↪URS0000000001
curl -H "Accept: application/yaml" https://rnacentral.org/api/v1/rna/
↪URS0000000001
```

(continues on next page)

(continued from previous page)

```
curl -H "Accept: text/html" https://rnacentral.org/api/v1/rna/
↪URS0000000001
```

If the Accept header and the format parameter conflict, the API returns HTTP status `406` with an error message, for example:

```
curl -H "Accept: application/json" "https://rnacentral.org/api/v1/rna/
↪URS0000000001/?format=yaml"
```

```
{
  "detail": "Could not satisfy the request Accept header."
}
```

Filtering

The API supports several filtering operations that complement the main RNAcentral *text search* functionality.

Filtering by sequence length

There are 3 url parameters: `length`, `min_length` (greater or equal length), and `max_length` (less or equal length).

Examples

- https://rnacentral.org/api/v1/rna/?length=2014
- https://rnacentral.org/api/v1/rna/?min_length=50000
- https://rnacentral.org/api/v1/rna/?max_length=10
- https://rnacentral.org/api/v1/rna/?min_length=10&max_length=100

Filtering by external ids

The external id is an id assigned to a sequence in one of the *Expert Databases*, which is imported into RNAcentral as a cross-reference.

The external id can be specified by setting the `external_id` url parameter.

Examples

- https://rnacentral.org/api/v1/rna/?external_id=Stap.epid._AF269831 (SRPDB)
- https://rnacentral.org/api/v1/rna/?external_id=MIMAT0000091 (miRBase)
- https://rnacentral.org/api/v1/rna/?external_id=OTTHUMG00000172092 (Vega)
- https://rnacentral.org/api/v1/rna/?external_id=Lepto_inter_Lai566 (tmRNA Website)

Filtering by MD5

Each sequence can be looked up by the MD5 hash of its uppercase DNA sequence using the `md5` url parameter, as in the *example script* below.

Example: https://rnacentral.org/api/v1/rna/?md5=06808191a979cc0b933265d9a9c213fd

Combined filters

Any filters can be combined to narrow down the query using the `&` symbol as a separator, which acts as the logical `AND` operator.

Example: https://rnacentral.org/api/v1/rna/?min_length=10&max_length=100

Genome annotations

The API provides an endpoint for retrieving annotations based on genomic coordinates for a *number of species*.

The response content is similar to that provided by the Ensembl REST API and can be directly used to visualise the data in a genome browser.

The genome location should be in the `chromosome:start-end` format and may contain optional commas.

Example: https://rnacentral.org/api/v1/overlap/region/homo_sapiens/2:39,745,816-39,826,679

Example script

A common task is finding whether some sequence of interest has an RNAcentral id.

An example **Python** script for **looking up RNAcentral unique sequence ids** (URS) is provided below and can be used in more complicated programs.

Instead of sending the entire sequence over the network to the RNAcentral servers, the script only submits the md5 value calculated based on the sequence. The md5 value is 32 characters long and is unique for each sequence. Using md5 saves traffic and decreases the response time.

Additional notes:

- all sequences are uppercase;
- md5 is computed for DNA sequences (so all Us should be replaced with Ts).

```
import hashlib
import requests  # pip install requests

def get_md5(sequence):
    """
    Calculate md5 for an RNA sequence.
    """
    # RNAcentral stores DNA md5 hashes
    sequence = sequence.replace('U', 'T')
    sequence = ''.join(map(str, sequence)).encode('ascii')

    # get the md5 digest
    m = hashlib.md5()
    m.update(sequence)
    return m.hexdigest()

def get_rnacentral_id(md5):
    """
    Parse json output and return the RNAcentral id.
```

(continues on next page)

(continued from previous page)

```
    """
    url = 'https://rnacentral.org/api/v1/rna'
    r = requests.get(url, params={'md5': md5})
    data = r.json()
    if len(data['results']) > 0:
        return data['results'][0]['rnacentral_id']
    else:
        return 'RNAcentral id not found'

# This sequence has an RNAcentral id
sequence =
↪'CUGAAUAAAUAAGGUAUCUUAUAUCUUUUAAUUAACAGUUAAACGCUUCCAUAAAGCUUUUAUCCA'
md5 = get_md5(sequence)
print(get_rnacentral_id(md5))  # URS00002C9E9D

# This sequence doesn't have an RNAcentral id
sequence = 'Not an RNA sequence'
md5 = get_md5(sequence)
print(get_rnacentral_id(md5))  # RNAcentral id not found
```

Cross-domain requests

Browsers only allow JavaScript on RNAcentral, EBI and a few partner sites to call the API directly. Web applications on other domains should request the data from their own server instead, or *get in touch* to have their domain added. JSONP is no longer supported.

Trailing slash

The trailing slash in all urls used in the API is **optional**, so https://rnacentral.org/api/v1/rna/URS00002C9E9D and https://rnacentral.org/api/v1/rna/URS00002C9E9D/ return the same entry.

2.14 Public Postgres database

In addition to *downloadable files*, an *API*, and the *text search*, RNAcentral provides a public Postgres database that can be used to query the data using SQL syntax. The database is updated with every RNAcentral release and contains a copy of the data available through the RNAcentral website.

2.14.1 Connection details

Main database

Setting	Value
Hostname	`hh-pgsql-public.ebi.ac.uk`
Port	`5432`
Database	`pfmegrnargs`
User	`reader`
Password	`NWDMCE5xdipIjRrp`

2.14.2 Connecting to the database

To connect to the database using the **command line**:

```
psql postgres://reader:NWDMCE5xdipIjRrp@hh-pgsql-public.ebi.ac.uk:5432/
↪pfmegrnargs
```

Pro tip: if you don't have `psql` installed on your machine, consider using Docker to get started with a pre-configured Postgres image:

```
docker pull postgres
docker run -it postgres psql postgres://reader:NWDMCE5xdipIjRrp@hh-pgsql-
↪public.ebi.ac.uk:5432/pfmegrnargs
```

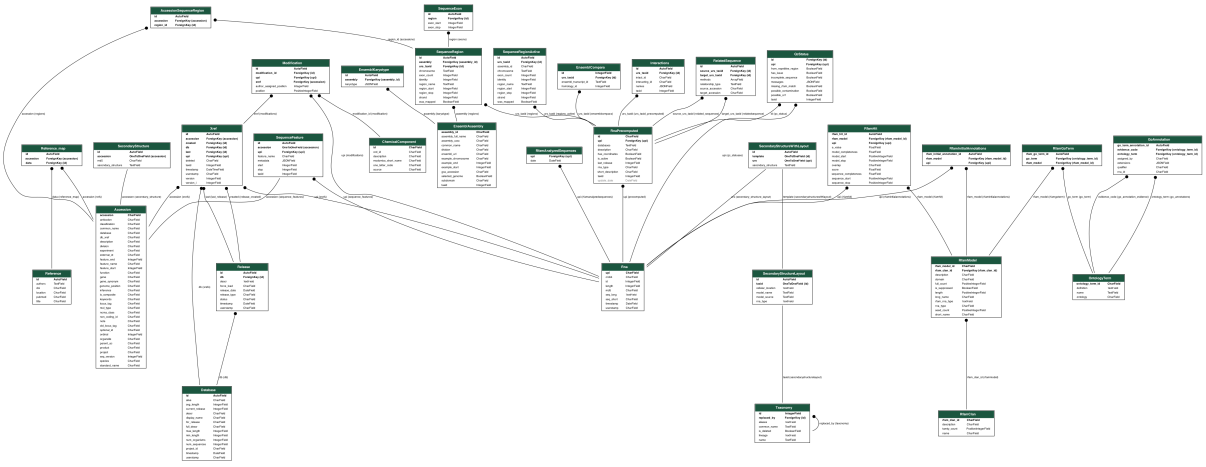
Alternatively, you can use a **Postgres client** like DBeaver or PgAdmin.

Warning

If your computer is behind a firewall, please ensure that outgoing TCP/IP connections to the corresponding ports are allowed.

2.14.3 Database schema

The following diagram was generated based on the latest version of RNAcentral.



2.14.4 Main tables

The entire RNAcentral schema contains more than 300 tables, but the following tables are good starting points for exploring the data:

Table	Description
`rna`	contains RNA sequences and *URS identifiers*
`xref`	contains cross-references to *Expert Databases*
`rnc_database`	contains a list of Expert Databases
`rnc_accessions`	contains metadata associated with each cross-reference
`rnc_rna_precomputed`	contains RNA types and descriptions for all sequences

2.14.5 Example queries

Search by external accessions

Although the VEGA database has been archived and its identifiers are no longer searchable using the RNAcentral text search, you can still query RNAcentral using VEGA identifiers:

```
SELECT
  urs,      -- RNAcentral URS identifier
  taxid,    -- NCBI taxid
  ac        -- external accession
FROM xref
WHERE ac IN ('OTTHUMT00000106564.1', 'OTTHUMT00000416802.1')
```

Example output:

```
URS00000B15DA        9606        OTTHUMT00000106564.1
URS00000A54A6        9606        OTTHUMT00000416802.1
```

2.14.6 Example workflow to extract all bacterial rRNA sequences

Exporting millions of sequences via the public database can be faster than using the *text search*. Here's an example of how you can explore this service:

1. Create a file `query.sql`:

   ```
   SELECT
       precomputed.urs_taxid
   FROM rnc_rna_precomputed precomputed
   JOIN rnc_taxonomy tax
   ON
       tax.id = precomputed.taxid
   WHERE
       tax.lineage LIKE 'cellular organisms; Bacteria; %'
       AND precomputed.is_active = true    -- exclude sequences without␣
   ↪active cross-references
       AND precomputed.rna_type = 'rRNA'
   ```

2. Run the following command to execute the query:

   ```
   docker run -v `pwd`:/rnacentral -it postgres /bin/sh -c 'cd /
   ↪rnacentral && psql -t -A -f query.sql postgres://
   ↪reader:NWDMCE5xdipIjRrp@hh-pgsql-public.ebi.ac.uk:5432/pfmegrnargs >␣
   ↪ids.txt'
   ```

   The command will create a file `ids.txt` with a list of RNAcentral identifiers.

3. Download the following RNAcentral FASTA file:

   https://ftp.ebi.ac.uk/pub/databases/RNAcentral/current_release/sequences/rnacentral_species_specific_ids.fasta.gz

4. Extract the sequences using seqkit:

```
seqkit grep -f ids.txt rnacentral_species_specific_ids.fasta.gz >␣
↪output.fasta
```

The file `output.fasta` will contain the desired subset of RNAcentral sequences in FASTA format.

2.14.7 Example Python script

Requires psycopg2 to connect to Postgres:

```
pip install psycopg2-binary
```

```
import psycopg2
import psycopg2.extras

def main():
    conn_string = (
        "host='hh-pgsql-public.ebi.ac.uk' dbname='pfmegrnargs' "
        "user='reader' password='NWDMCE5xdipIjRrp'"
    )
    conn = psycopg2.connect(conn_string)
    cursor = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)

    # retrieve a list of RNAcentral databases
    query = "SELECT * FROM rnc_database"

    cursor.execute(query)
    for row in cursor:
        print(row)

if __name__ == "__main__":
    main()
```

2.15 Bulk downloads

RNAcentral data can be downloaded in bulk from the FTP Archive, queried through the REST API or the public Postgres database, or exported from the text search.

2.15.1 Choosing an access method

Task	Recommended access method
Download complete release files	*FTP Archive*
Download archived release snapshots	*FTP Archive*
Retrieve one sequence or a small result set	*REST API*
Integrate RNAcentral into software	*REST API*
Map many identifiers in bulk	*Identifier mapping files*
Run SQL queries across RNAcentral tables	*Public Postgres database*
Export a filtered subset of sequences	*Text search export*

2.15.2 FTP Archive

To retrieve **large amounts of data** or to download **archived data snapshots**, use the RNAcentral FTP Archive at https://ftp.ebi.ac.uk/pub/databases/RNAcentral (no authentication required).

The files are produced during each RNAcentral release. The current release is in current_release, and every release back to 1.0-beta is kept in releases. Older releases contain largely the same files, though the data has evolved over time.

Most files are compressed with gzip and end in `.gz`; decompress them with `gzip -d <file path>`. Most directories also contain a readme file that explains their contents.

Files available

Name	Description	Format	Link
Database files	A dump of the Postgres database, in directory format (`toc.dat` plus one file per table), which can be restored with pg_restore. Removed from older releases.	pg_du	database_files
Genome coordinates	Coordinates of RNAs in RNAcentral in each model organism, as annotated by expert databases	BED, GFF3	BED, GFF3
GO annotations	Mappings of RNAcentral entries to Gene Ontology terms	TSV	rnacentral_rfam_annotations.tsv.gz
GPI	Gene product information for selected RNAs, with separate species and reference proteome files	GPI	gpi
ID mapping	Mapping of RNAcentral IDs to expert database IDs, also available per database	TSV	id_mapping.tsv.gz, per database
JSON	RNAcentral IDs and their cross-references to Ensembl, 100,000 sequences per file	JSON	json
MD5	RNAcentral IDs mapped to the MD5 hash of each sequence	TSV	md5.tsv.gz
Rfam	RNAcentral IDs with their associated Rfam annotations	TSV	rfam_annotations.tsv.gz
Sequences (active)	Sequences present in at least one expert database	FAST	rnacentral_active.fasta.gz
Sequences (inactive)	Sequences no longer present in any expert database	FAST	rnacentral_inactive.fasta.gz
Sequences (species specific)	Active sequences with species-specific IDs (`URS…_taxid`)	FAST	rnacentral_species_specific_ids.fasta.gz
Sequences (by database)	Active sequences, one file per expert database	FAST	by-database
Sequences (by species)	Species-specific sequences, one file per species, grouped by taxonomic group	FAST	by-species

Identifier mapping files

Use the ID mapping files when you need to map many identifiers between RNAcentral and *expert databases*:

- combined file: `current_release/id_mapping/id_mapping.tsv.gz`
- per-database files: `current_release/id_mapping/database_mappings/`

See *Linking to RNAcentral* for more on identifier mapping.

Directory structure

```
RNAcentral
|
+- current_release
|   |
|   +- release_notes.txt
|   |
|   +- database_files
|   |   |
|   |   +- toc.dat and one gzip compressed file per table
|   |
|   +- genome_coordinates
|   |   |
|   |   +- bed
|   |   |   |
|   |   |   +- one gzip compressed BED file per model organism
|   |   |
|   |   +- gff3
|   |   |   |
|   |   |   +- one gzip compressed GFF3 file per model organism
|   |   |
|   |   +- readme.txt
|   |
|   +- go_annotations
|   |   |
|   |   +- rnacentral_rfam_annotations.tsv.gz
|   |
|   +- gpi
|   |   |
|   |   +- rnacentral.gpi(.gz)
|   |   |
|   |   +- rnacentral.species.gpi(.gz)
|   |   |
|   |   +- rnacentral.reference_proteome.gpi(.gz)
|   |
|   +- id_mapping
|   |   |
|   |   +- database_mappings
|   |   |   |
|   |   |   +- one uncompressed TSV file per expert database
|   |   |
|   |   +- id_mapping.tsv.gz
```

(continues on next page)

(continued from previous page)

```
|   |
|   +- json
|   |   |
|   |   +- ensembl-xref-*.json
|   |
|   +- md5
|   |   |
|   |   +- md5.tsv.gz
|   |
|   +- rfam
|   |   |
|   |   +- rfam_annotations.tsv.gz
|   |
|   +- sequences
|       |
|       +- by-database
|       |   |
|       |   +- one gzip compressed FASTA file per expert database
|       |
|       +- by-species
|       |   |
|       |   +- one folder per taxonomic group, one gzip compressed FASTA␣
↪file per species
|       |
|       +- rnacentral_active.fasta.gz
|       |
|       +- rnacentral_inactive.fasta.gz
|       |
|       +- rnacentral_species_specific_ids.fasta.gz
|
+- releases
    |
    +- archive of releases back to 1.0-beta (2014)
```

2.16 RNAcentral MCP server

The **RNAcentral MCP server** exposes the RNAcentral database to AI assistants through the Model Context Protocol (MCP). It lets MCP-capable LLM clients, such as Claude Desktop and Claude Code, answer questions about non-coding RNA by calling RNAcentral directly, instead of you chaining together raw API requests.

Behind the scenes the server orchestrates several RNAcentral services (EBI Search, the *RNAcentral REST API*, Ensembl, and the *sequence search* and export microservices) so that a single question can be resolved with a single tool call.

2.16.1 What you can do with it

The server registers a set of tools that let an LLM:

- **Search RNAcentral** by text or metadata, with enriched results including Rfam hits and GO annotations
- **Run sequence similarity searches** from a raw RNA sequence
- **Map identifiers** between *RNAcentral URS IDs* and external database IDs
- **Export sequences** in bulk (FASTA or Parquet)
- **Find overlapping ncRNAs** for a set of genomic coordinates or a gene
- **Retrieve *secondary structure* diagrams** for a given URS ID
- **Get *literature summaries*** describing an RNA

This makes it easy to ask natural-language questions such as *"find ncRNAs overlapping this GWAS locus"*, *"build a training set of tRNA sequences"*, or *"show me the 2D structure and literature summary for this RNA"* and have the assistant gather the answer for you.

2.16.2 Responsible use

The MCP server is the preferred interface for AI assistants carrying out RNAcentral research tasks. For direct API calls, bulk retrieval, or automated workflows, follow the *responsible automated access guidance*, including its rate limits and retry advice.

2.16.3 Quick start

The server is published on PyPI as `rnacentral-mcp-server` and requires **Python 3.10+**. With uv it can be fetched and run on demand with no explicit install step:

```
uvx --from rnacentral-mcp-server run-server
```

Register that command with your MCP client. For example, in Claude Code:

```
claude mcp add rnacentral -- uvx --from rnacentral-mcp-server run-server
```

2.16.4 Full documentation

Installation guides for Claude Desktop, Claude Code, and other MCP clients, together with the per-tool reference and examples, are maintained separately:

- **Documentation:** https://rnacentral.github.io/rnacentral-mcp-server/
- **Glama MCP directory listing:** https://glama.ai/mcp/servers/RNAcentral/rnacentral-mcp-server (hosted install, schema browser, and a step-by-step usage guide)
- **Source code:** https://github.com/RNAcentral/rnacentral-mcp-server
- **Package:** https://pypi.org/project/rnacentral-mcp-server/

2.17 Genome locations

2.17.1 About genomic mapping

RNAcentral imports genome locations from Ensembl, LNCipedia, miRBase, and other *Expert Databases*. In addition, sequences are mapped to the reference genomes downloaded from **Ensembl** and **Ensembl Genomes** using blat.

Sequences are mapped to reference genomes from **more than 250 species**.

2.17.2 Genome browser

RNAcentral sequences can be viewed in their genomic context using a **genome browser** powered by IGV: https://rnacentral.org/genome-browser

IGV offers various navigation and zooming features, allowing users to explore genomic regions by clicking and dragging with the mouse, utilizing zoom buttons, or entering specific coordinates or a gene name in the search box for quick navigation. It also provides extra customization options, including the ability to modify color schemes, track configurations (heights, visibility, etc.), and export views in SVG format.

The genome browser is shown in the **Genome locations** section of each sequence page, opened at the location of that RNA. The RNA in question is highlighted in yellow and shown at the top of the RNAcentral track, and the **Ensembl annotation** track above it shows the protein-coding and non-coding genes annotated by Ensembl at the same location, for comparison.

RNA modification sites

For human (GRCh38) and mouse (GRCm39), the genome browser can also show RNA modification sites from Sci-ModoM, a quantitative database of transcriptome-wide, high-throughput RNA modification data. Tick **SciModoM** under **Track hubs** to add the track. Each modification type, such as m6A or m5C, is plotted in a different colour, and each site reports a confidence score, the read coverage and the percentage of reads that are modified.

Sharing a view

The address bar updates as you move around the genome browser, so you can copy the URL to bookmark or share the region you are looking at. To build links to genome locations from your own website or tool, see *Linking to genome locations*.

2.17.3 Finding mapped sequences

Sequences with genomic coordinates can be found with the text search field `has_genomic_coordinates:"True"` (see *Text search*).

2.17.4 Downloading coordinates

The genomic coordinates of the RNAcentral entries can be **downloaded** in BED and GFF3 formats from the *FTP Archive* or retrieved through the *REST API*.

2.18 Genes

RNAcentral groups related sequences into **genes**. Each RNAcentral sequence is a single transcript, and many of them come from the same locus, for example the dozens of lncRNA transcripts of a single lncRNA gene imported from different *Expert Databases*. A gene brings these transcripts together in one place.

Genes are available for more than 1.6 million loci across the species annotated by Ensembl.

2.18.1 Gene identifiers

Each gene has an RNAcentral gene identifier such as `RNACG65104808090.4`, made up of a prefix, a number, and a version. The prefix is `RNAC` followed by the species' Ensembl prefix and `G`, for example `RNACG` for human, `RNACMUSG` for mouse and `RNACDARG` for zebrafish. When a gene in a new release covers the same region as a gene in the previous release (or almost the same, with more than 90% overlap), it keeps that gene's identifier with a new version number. Genes that are no longer found are retired, and new genes get new identifiers.

Genes are separate from the *URS identifiers* of the sequences they contain, which stay the same whatever gene a sequence belongs to.

2.18.2 How genes are built

Genes are computed with every release from the *genome locations* of RNAcentral sequences. A machine learning model compares pairs of transcripts at the same locus, using features such as how much their exons overlap, their strands, and how similar their RNA types are, and predicts whether they belong to the same gene. Transcripts that are linked in this way are then grouped into genes. Only groups of at least two transcripts become genes.

2.18.3 Gene pages

Each gene has its own page, for example the human *RN7SL1* gene RNACG65104808090.4, showing:

- a **gene summary** with the gene type, chromosome, start and end positions, strand and length, and a link to the *genome browser*;
- the **transcripts**, the RNAcentral sequences that make up the gene;
- an *AI-generated literature summary*, where one is available;
- **external links** to other resources.

2.18.4 Finding genes

Genes appear in the *text search* results alongside sequences. To search only for genes, use `entry_type:"Gene"`. Search results that contain only genes can't be exported in FASTA format, as genes don't have a single sequence.

2.19 Secondary structure

RNAcentral generates secondary structure (2D) diagrams using the R2DT software that visualises RNA structure using standard layouts or templates. Learn more about the method in the R2DT paper or check out the code on GitHub.

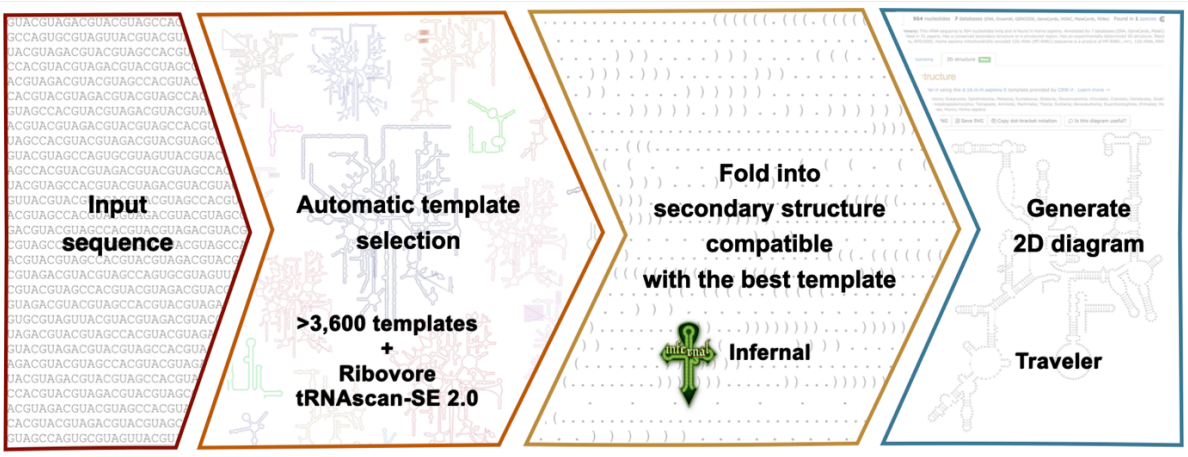
2.19.1 R2DT overview

RNAcentral stores RNA secondary structures computed using R2DT for millions of non-coding RNAs. The structures are displayed using a library of ~4,000 templates, including:

1. Small subunit (SSU) rRNA and 5S rRNA templates from the Comparative RNA Website (CRW);
2. Small and large subunit (LSU) rRNA templates from RiboVision;

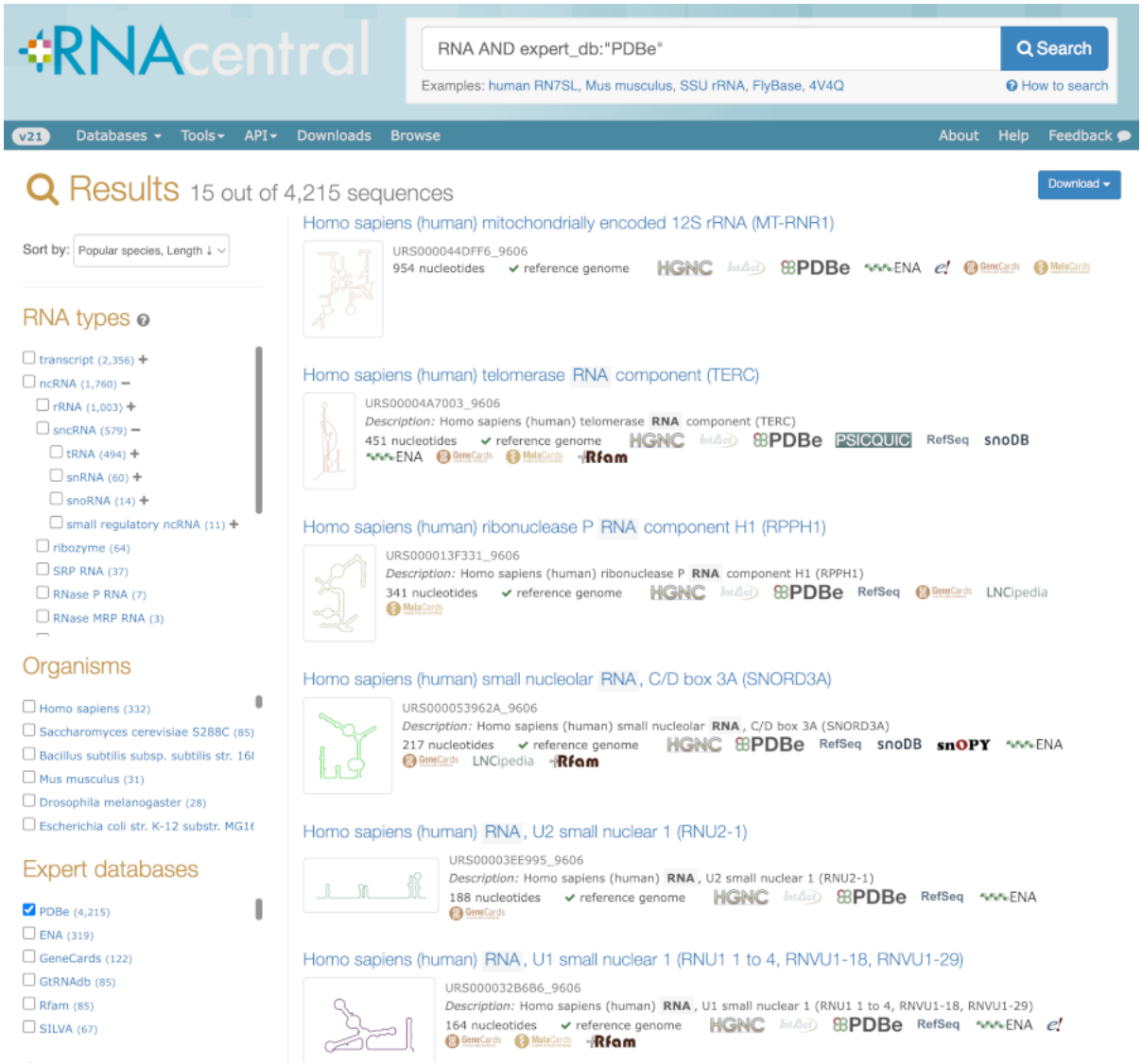
3. Isotype-specific tRNA templates from GtRNAdb;
4. RNase P templates from the RNase P database;
5. Rfam consensus secondary structures for all other RNA types.

For each sequence the **R2DT pipeline** automatically selects the template using ribovore, folds the sequence into the template structure using Infernal, and visualises it using Traveler.

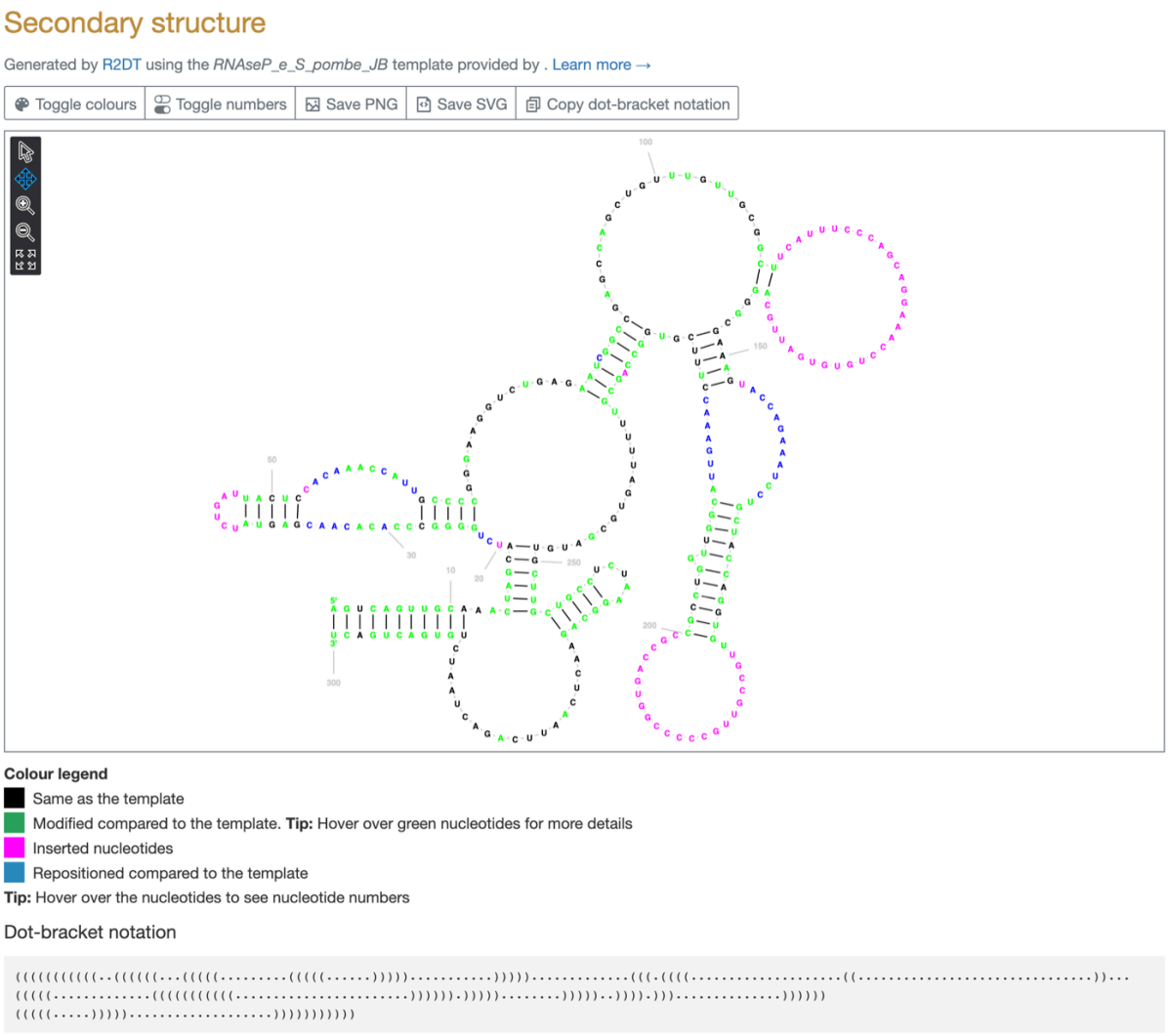


2.19.2 RNA secondary structure in RNAcentral

The secondary structures are displayed in text search results, on RNA sequence pages, and in sequence similarity search results. For example, text search results include simplified outlines of the structures:



Clicking on any thumbnail shows a detailed diagram, such as the following fly RNase P structure:



What do colours mean?

The differences between the template and the sequence are highlighted in colour. The nucleotides shown in black are identical to the template while the nucleotides that differ from the template are shown in several colours depending on whether it's an insertion (magenta) or modification (green).

2.19.3 Running R2DT yourself

To draw the secondary structure of your own sequences, use the R2DT web application. R2DT can also be run through an API or on the command line. See *R2DT* under Tools.

Browse all sequences with 2D structures →

2.20 QC annotations

RNAcentral provides a variety of quality checks for all sequences. Many of these checks are based on Rfam. Rfam is a database of functional non-coding RNA families represented by multiple sequence alignments and consensus secondary structures. The sequence and structural information is used to build Infernal covariance models, which can be used to find new instances of RNA families and annotate genomes with non-coding RNAs.

RNAcentral annotates **all sequences** with Rfam models for each release. Rfam classification provides

additional context to sequences with few annotations and helps identify **potential problems**, for example, sequences which are likely contamination.

In addition, three tools are used to detect potential protein-coding regions (see *Possible ORFs*).

2.20.1 Types of quality control

1. Incomplete sequences

When an RNAcentral sequence matches only a **part of the Rfam covariance model**, the sequence is flagged as incomplete.

Examples

- Partial SSU rRNA sequence
- Partial eukaryotic LSU sequence
- Partial Bacterial SSU sequence

These sequences can be browsed by searching for `qc_warning:"incomplete_sequence"`.

2. Potential contamination

For example, when an **Eukaryotic sequence** matches an Rfam family that is only found in **Bacteria**, this could indicate bacterial contamination or taxonomic misclassification.

Examples

- Mouse rRNA matching Bacterial SSU model
- Archaeal RNA matching Bacterial SRP

Sequences of this type can be browsed with `qc_warning:"possible_contamination"`.

3. Missing Rfam hits

The majority of RNAcentral sequences annotated as rRNA or tRNA match the corresponding Rfam families. However, some sequences do not match the expected Rfam families which could mean that either the sequence has an incorrect RNA type or that the Rfam model needs to be updated.

Examples

- Sequence annotated as tRNA that matches Group II intron
- Sequence annotated as rRNA that matches a riboswitch
- Sequence annotated as tRNA that matches SSU rRNA

Browse all sequences with missing matches by searching for `qc_warning:"missing_rfam_match"`.

4. Possible ORFs

Very few sequences contain open reading frames, however, it is worth noting which do as this may influence the function of the ncRNA. Three tools are used to detect them, and each has its own warning:

- CPAT (v3.0.4, default options) detects possible open reading frames and is run on all human, fly, mouse, and zebrafish sequences (`qc_warning:"possible_orf"`).

- tcode (EMBOSS v6.6.0 with default options) applies the Fickett TESTCODE statistic and is run across all organisms on sequences longer than 200 nt (`qc_warning:"possible_orf_tcode"`).
- stopFree, an in-house Rust implementation of the tool described in Evaluating computational tools for protein-coding sequence detection: Are they up to the task? (v0.2.4), measures the longest stop-codon-free run in each reading frame and is run on all sequences (`qc_warning:"possible_orf_stopfree"`).

Examples

- Human sRNA
- Fly lncRNA

2.20.2 Quality control table

The QC status section of each sequence page has a summary table for assessing the status of a sequence at a glance. The table reports the results of coding potential, contamination, and Rfam quality-control checks, using a green tick to indicate a passing result and a red cross to indicate a failed result.

For the coding potential row, you may also see an amber status. This row can be expanded to view the results of the three coding potential tests currently used: CPAT, tcode, and stopFree. Each individual test result is displayed as either pass (green), ORF predicted (red), or not tested (grey).

When all three tests agree, the overall coding potential status is shown as either green or red. However, when the tests produce conflicting results, the overall status is shown in amber to indicate disagreement between the methods.

- Example of sequence with an ORF called by all three coding potential tools: Human circRNA
- Example of sequence with a disagreement between methods: Human lncRNA

2.20.3 Why some sequences do not match any Rfam families

There are several possible reasons:

- **not all RNA types are represented in Rfam**

  For example, **piRNAs or mature miRNAs** are too short to be accurately modelled in Rfam. Conversely, **lncRNAs** tend to be too long and poorly conserved, although Rfam includes several conserved lncRNA domains.

- this family may not yet exist in Rfam
- the Rfam model needs to be updated to include this sequence

If you would like to suggest a new Rfam family or report an error, please *get in touch* using the **Feedback** button found at the top right of every RNAcentral page.

2.21 Gene Ontology annotations

2.21.1 What are GO terms?

Gene Ontology (GO) terms provide computer-readable annotations capturing biological processes, molecular functions, and cellular components associated with genes and transcripts. For more information about Gene Ontology, see Introduction to GO or Ten Quick Tips for Using the Gene Ontology in PLoS.

2.21.2 How are GO terms added to RNAcentral sequences?

RNAcentral sequences are **manually annotated** with GO terms by several teams, including BHF-UCL, ARUK-UCL, MGI, FlyBase and SGD. Find out more about literature curation using RNAcentral identifiers in the following paper:

> **Guidelines for the functional annotation of microRNAs using the Gene Ontology** *Huntley et al., RNA 2016.* DOI: 10.1261/rna.055301.115

In addition, RNAcentral sequences are **automatically** annotated with GO terms propagated from the matching Rfam covariance models. The process is described in GO_REF:0000115.

GOFlowLLM

Some miRNA annotations are curated from open-access papers by **GOFlowLLM**, a large language model that follows the GO consortium's miRNA curation guidelines step by step. Its annotations agree with expert GO curators in around 90% of cases. Sequences annotated this way have a **GOFlowLLM** tab showing the evidence from the paper and each decision the model made. A curated paper may still contain information that GOFlowLLM does not extract, as it only follows these guidelines.

Find out more on the *GOFlowLLM* page, or browse the annotated sequences.

2.21.3 Example annotations

- miRNA involved in heart development
- yeast snoRNA
- Mouse KCNQ1 lncRNA

Browse all sequences with GO annotations →

2.21.4 Searching GO terms

The GO annotations are searchable using the *text search*, for example:

- Find all sequences with annotations: has_go_annotations:"True"
- Find all sequences annotated by BHF-UCL: go_annotation_source:"BHF-UCL"
- Find all sequences annotated by ARUK-UCL: go_annotation_source:"ARUK-UCL"
- Search by qualifier, for example: involved_in:"GO:0043410", involved_in:"GO:2000352", enables:"GO:0005515" or by searching with the name of the GO term like enables:"protein binding".

The following qualifiers are supported: `part_of`, `involved_in`, `enables`, `contributes_to`, and `colocalizes_with`.

At this time the search is not ontology-aware, so only the GO terms from the query are used for searching, not their descendants or ancestors.

2.21.5 Downloading GO annotations

The GO annotations can be downloaded or browsed in **QuickGO**. GO annotation files are also included in the *FTP Archive*.

Browse annotations in QuickGO →

If you notice a problem with GO terms, please *get in touch* using the **Feedback** button found at the top right of every RNAcentral page.

2.22 RNA-target interactions

RNAcentral shows experimentally supported interactions between RNAs and their targets from three *Expert Databases*: miRNA targets from TarBase and LncBase, and curated RNA interactions from IntAct. Further relationships from the RNA-KG knowledge graph are shown in the **Relationships** tab.

2.22.1 miRNAs targeting mRNAs

miRNAs regulate gene expression of protein-coding genes by interacting with target mRNAs. Hundreds of thousands of experimentally supported miRNA:gene pairs are cataloged in TarBase and can be viewed and searched in RNAcentral.

- Browse human miRNAs with protein targets
- Browse mouse miRNAs with protein targets

2.22.2 miRNAs targeting lncRNAs

RNAcentral also imports information about experimentally confirmed miRNAs that target lncRNAs from LncBase.

- Browse human miRNAs with lncRNA targets
- Browse mouse miRNAs with lncRNA targets

2.22.3 Interactions curated by IntAct

IntAct curates molecular interactions from the literature, including interactions of around 1,400 RNAs with proteins and other RNAs. Most are human, with others from yeast, mouse and rat, and they cover miRNAs, snoRNAs, lncRNAs and other RNA types. Each interaction links to the partner protein in UniProt or the partner RNA in RNAcentral, and to the evidence in IntAct.

- Browse RNAs with interactions from IntAct

2.22.4 Interactions on sequence pages

- On miRNA pages you can see protein and lncRNA targets, for example hsa-miR-21-5p.
- On lncRNA pages, a **Targeting miRNAs** section lists the miRNAs that target the lncRNA, according to LncBase.
- Interactions from IntAct are listed with a **View in IntAct** link, for example for the GAS5 lncRNA.

2.22.5 Molecular relationships

The **Relationships** tab of a sequence page lists relationships between the RNA and other biological entities from RNA-KG, a knowledge graph built from many sources. Depending on the RNA, these can include proteins it interacts with or regulates, other RNAs, genetic variants, GO terms, diseases and drugs. Each relationship shows its type, the source, the experimental method and the supporting publications, and the list can be filtered by relationship type and source.

For example, the hsa-mir-126 precursor has several hundred relationships, including:

Related entity	Example	Source
RNA	interacts with the GAS5 lncRNA	RNAInter
Protein	interacts with SMC3	RNAInter
Genetic variant	causally influenced by the somatic variant COSV99056770	COSMIC
Disease	somatic mutations linked to squamous cell lung carcinoma	SomamiR
GO term	functions in the execution phase of apoptosis	TAM
Drug or chemical	interacts with insulin	miRNet

2.22.6 Searching interactions

- Find miRNAs that target a specific protein: interacting_protein:"ENSG00000277791"
- Find miRNAs that target a specific lncRNA: interacting_rna:"ENSG00000235652"
- Browse miRNA targets by experimental method. Around 40 methods are recorded; the most common include:
  - evidence_for_interaction:"HITS-CLIP"
  - evidence_for_interaction:"PAR-CLIP"
  - evidence_for_interaction:"qCLASH"
  - evidence_for_interaction:"Luciferase Reporter Assay"
  - evidence_for_interaction:"Western Blot"
  - evidence_for_interaction:"qPCR"
  - evidence_for_interaction:"CLASH"
  - evidence_for_interaction:"Microarrays"

The `has_interacting_proteins`, `has_interacting_rnas` and `evidence_for_interaction` searches cover TarBase and LncBase. To find RNAs with interactions from IntAct, search for `expert_db:"IntAct"`.

2.23 Expression and localisation

Sequence pages show where an RNA acts: where in the body it is expressed, and where in the cell it is found.

2.23.1 Gene expression

RNAcentral shows gene expression data from Expression Atlas, EMBL-EBI's resource of gene and protein expression across species and biological conditions such as tissues, cell types, developmental stages and diseases.

Around 260,000 RNAcentral sequences are linked to Expression Atlas, mostly human and mouse lncRNAs.

When a sequence is linked to Expression Atlas and has an Ensembl gene, its sequence page has an **Expression** tab. The tab shows an Expression Atlas heatmap of the expression of that gene across the experiments and conditions available in Expression Atlas, so you can see, for example, which tissues and organs the RNA is expressed in. Links lead to the full results in Expression Atlas.

For example, see the Expression tab for human XIST.

Sequences linked to Expression Atlas can be found with the text search query `expert_db:"Expression Atlas"` (see *Text search*).

2.23.2 Subcellular localisation

When an RNA has a *Gene Ontology* annotation saying it is part of a cellular component, such as the nucleus or the cytoplasm, the **Localisation** section of its sequence page shows a SwissBioPics cell diagram with those components highlighted. For example, see hsa-miR-138-5p, which is found in the nucleus and the cytoplasm.

2.24 Taxonomy and homologs

Sequence pages show where else an RNA is found: in which species the same sequence occurs, and which related RNAs exist in other species.

2.24.1 Taxonomy

The same sequence can be found in more than one species, in which case it has a single *URS identifier* and a species-specific identifier for each species. The **Taxonomy** tab of a sequence page shows a taxonomic tree of all the species in which the sequence is found, and lists the entry for each of them. Click a species name to view its annotations, and click tree nodes to collapse or expand them.

For example, see the Taxonomy tab for hsa-miR-21-5p, which is found in 30 species.

2.24.2 Orthologs and paralogs

For sequences from Ensembl, the **Ensembl Compara** section of the sequence page lists the RNA's orthologs and paralogs, related RNAs in other species and in the same species, from the Ensembl Compara gene trees. Paralogs are labelled. This information is only available for some RNA types annotated by Rfam.

For example, see the hsa-mir-126 precursor, which has more than 100 orthologs and paralogs.

2.25 Sequence features

Each sequence page has a **Feature viewer** that shows annotations along the sequence. Each type of feature has its own track, and only tracks with data for that sequence are shown. Hover over a feature to see its details.

2.25.1 Rfam families

Regions of the sequence that match Rfam families, from the Rfam classification that RNAcentral runs on every sequence (see *QC annotations*).

2.25.2 Conserved features

Conserved RNA motifs or conserved RNA structures (CRS) are structured elements of RNA molecules that are conserved across species. They have been computationally identified in multi-species vertebrate alignments from **structure-based**, rather than sequence-based, alignments in the following study:

> **The identification and functional annotation of RNA structures conserved in vertebrates**
> *Seemann et al., Genome Research 2017.* DOI: 10.1101/gr.208652.116

RNAcentral collected non-overlapping CRS motifs that were additionally filtered for Rfam overlap, and links each one to the RTH Database, where additional information can be found, including multiple sequence alignments, secondary structure, and tissue expression.

Examples:

- Human prostate cancer associated non-coding RNA 1 (PRNCR1)
- Mouse eosinophil granule ontogeny transcript (Egot, Gm27813)

Sequences with conserved features can be found with `has_conserved_structure:"True"`.

2.25.3 Mature miRNA

The positions of the mature miRNAs within a miRNA precursor, from miRBase. Clicking a mature miRNA opens it in miRBase. For example, see the hsa-mir-126 precursor.

2.25.4 CPAT ORF

Open reading frames predicted by CPAT, with the coding probability and the cutoff used. CPAT is run on human, mouse, fly and zebrafish sequences (see *Possible ORFs*). For example, see this human circRNA.

Sequences with a CPAT ORF can be found with `qc_warning:"possible_orf"`.

2.25.5 RNA editing events

RNA editing sites from REDIportal, with the edit and the reference nucleotide. For example, see this human lncRNA.

Sequences with editing events can be found with `has_editing_event:"True"`.

2.25.6 Anticodon

The anticodon of a tRNA, with its isotype. For example, see human tRNA-His-GTG-2-1.

2.25.7 Non-canonical

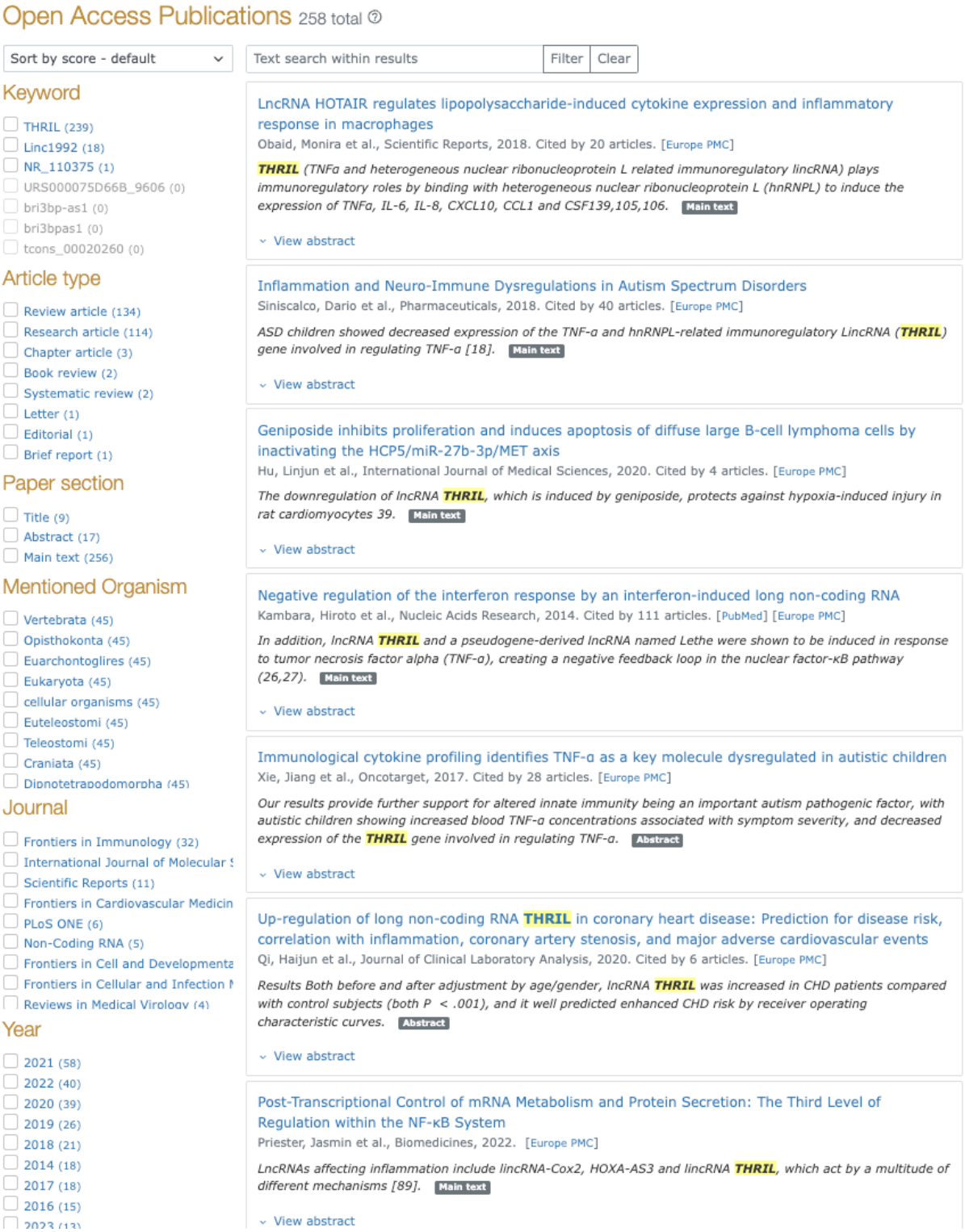
Positions in the sequence that are not A, U, G or C, such as the ambiguity codes N (any nucleotide) or Y (C or U). For example, see this 5S rRNA, which starts with two Y nucleotides.

2.26 LitScan

LitScan is a text mining pipeline that connects RNA sequences with the latest open access scientific literature. LitScan uses a collection of identifiers (Ids), gene names, and synonyms provided to RNAcentral by the *Expert Databases* to scan the papers available in Europe PMC, and is updated with every RNAcentral release.

The papers are listed in the **Publications** tab of each sequence page, where they can be filtered by identifier, article type, the section of the paper where the Id is found, mentioned organism, journal, and year.

For example, lncRNA `THRIL` is also known as `Linc1992`. Using LitScan, the corresponding RNAcentral entry includes papers about `THRIL`, `Linc1992`, and even `NR_110375` which is another Id for the same gene:



Browse all sequences with publications →

2.26.1 How LitScan works

With every release, LitScan:

1. **Collects the Ids** (identifiers, gene names and synonyms) of RNAcentral sequences from the Expert Databases. Ids that are numbers, common English words such as `hairpin`, `nail`, `digit` or `eric`, or too short to be specific are left out to prevent false matches.

2. **Searches Europe PMC** for open access articles that mention each Id. New Ids are searched across all of Europe PMC, and Ids that were searched before are checked for articles published since the last search. The query is:

```
"<id>" AND ("rna" OR "mrna" OR "ncrna" OR "lncrna" OR "rrna" OR "sncrna
↪") AND IN_EPMC:Y AND OPEN_ACCESS:Y AND NOT SRC:PPR
```

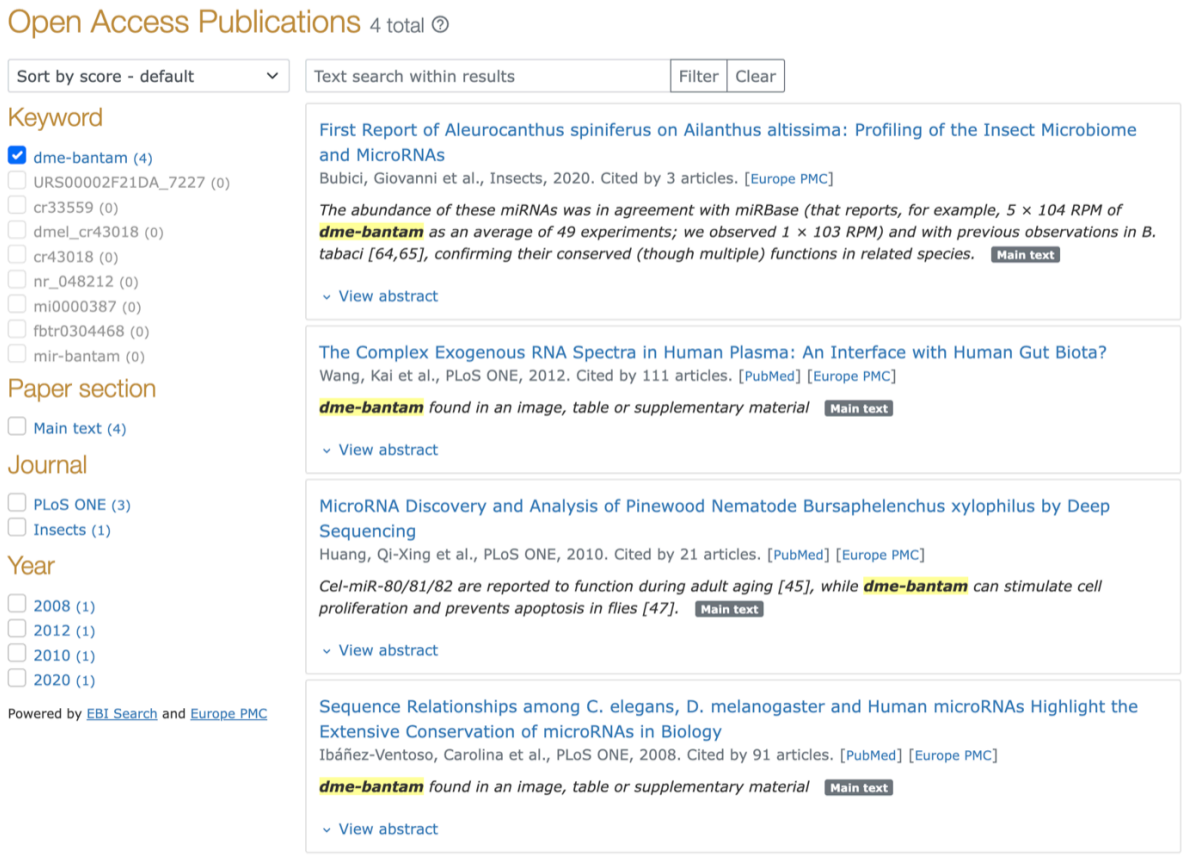
   where the RNA terms filter out possible false positives, `IN_EPMC:Y` and `OPEN_ACCESS:Y` require the full text to be openly available in Europe PMC, and `NOT SRC:PPR` excludes preprints, which are not peer-reviewed.

3. **Confirms the exact Id** in the full text of each article. Europe PMC's search treats spaces and special characters as separators, so it can return articles that only contain part of an Id. LitScan uses regular expressions to find the exact Id in the title, abstract, or body of the article, and extracts the sentences that mention it. Articles where the exact Id is not found are not shown.

4. **Annotates the articles.** Retracted articles are removed, articles that Expert Databases have curated for an RNA are marked, and the organisms mentioned in each article are identified using the ORGANISMS project (paper), so they may not be entirely accurate.

Example

A search on EuropePMC for the microRNA precursor dme-bantam ID yielded 13 results as of October 2023. However, the second step finds the exact string `dme-bantam` **only in 4 articles**, while the other 9 mention dme-bantam-3p and/or dme-bantam-5p and appear on the corresponding mature microRNA pages.



2.26.2 Use LitScan in your website

The LitScan widget is implemented as an **embeddable component** that can be used by any *Expert Database* or any other website. LitScan has already been deployed on the Rfam website (for example, see the SAM riboswitch page).

Find out more about how to integrate this widget into your website.

2.26.3 Finding the Ids used by LitScan

The list of Ids searched for each Expert Database can be accessed programmatically, for example with the following Python script. EBI Search returns at most 100,000 results for a query, so for the largest databases (such as Ensembl, RefSeq and miRBase) the script retrieves only the first 100,000 entries.

```
"""
Usage: python get_ids.py [database]
Example: python get_ids.py rfam
"""

import itertools
import json
import math
import requests  # pip install requests
import sys

# EBI Search returns at most the first 100,000 results of a query
MAX_RESULTS = 100000


def get_query(database, start):
    """
    Function to build the query
    :param database: name of the Expert DB
    :param start: index of the first entry to fetch
    :return: query
    """
    query = "?query=entry_type:metadata%20AND%20database:" \
            + database \
            + "&fields=job_id&size=100&start=" \
            + str(start) \
            + "&format=json"
    return query


def get_results(database):
    """
    Function to get the list of ids from a given database
    :param database: name of the Expert DB
    :return: results
    """
    start = 0
    endpoint = "https://www.ebi.ac.uk/ebisearch/ws/rest/rnacentral-litscan"
```

(continues on next page)

(continued from previous page)

```
    query = get_query(database, start)
    db_request = requests.get(endpoint + query)
    hit_count = json.loads(db_request.text)["hitCount"]
    results = []

    if hit_count > 0:
        # get ids
        get_result = json.loads(db_request.text)
        entries = [get_result["entries"]]
        results.append([item["fields"]["job_id"][0] for sublist in entries␣
↪for item in sublist])

        if hit_count > MAX_RESULTS:
            print("{} has {} entries; only the first {} can be retrieved".
↪format(
                database, hit_count, MAX_RESULTS))

        # fetch other ids if needed
        iter_number = int(math.ceil(min(hit_count, MAX_RESULTS) / 100.0))
        for num in range(iter_number - 1):
            start += 100
            query = get_query(database, start)
            new_request = requests.get(endpoint + query)
            new_request_result = json.loads(new_request.text)
            new_entries = [new_request_result["entries"]]
            results.append([item["fields"]["job_id"][0] for sublist in new_
↪entries for item in sublist])

        return results


def main():
    database = None

    if len(sys.argv) == 1:
        print("You must specify the database")
        exit()
    elif len(sys.argv) == 2:
        database = sys.argv[1]
    else:
        print("Usage: python get_ids.py rfam")
        exit()

    if database:
        # get list of ids
        results = get_results(database)

        if results:
            results = list(itertools.chain.from_iterable(results))
```

(continues on next page)

(continued from previous page)

```
            results = sorted(set(results))

            # save results
            with open(database + '_ids.txt', 'w') as f:
                for item in results:
                    f.write(item + '\n')
        else:
            print("No id found for database {}".format(database))


if __name__ == "__main__":
    main()
```

2.27 LitSumm

LitSumm uses large language models to write short summaries of what is known about non-coding RNAs from the scientific literature. Each summary is built from the open access articles that *LitScan* links to the RNA, and cites the articles it draws on. The approach is described in LitSumm: large language models for literature summarization of noncoding RNAs.

Browse all sequences with LitSumm summaries →

2.27.1 Summaries on RNAcentral pages

Summaries appear near the top of a sequence page, below the RNA type, under an **AI-generated content** notice. A sequence can have more than one summary, one for each name it is known by in the literature. Each article cited in a summary is linked to Europe PMC. For example, see the summary for human MIR944.

Gene pages show the summaries of the transcripts in the gene.

2.27.2 How summaries are made

For each RNA, a language model reads the relevant passages of the articles found by LitScan and writes a summary that cites each statement. The summary is then checked automatically against the articles and revised where needed.

2.27.3 Limitations

Summaries are generated automatically and may not be completely accurate or comprehensive. Known issues include:

- missing or incorrectly formatted references;
- similar statements from different articles merged incorrectly;
- inaccurate statements.

If you notice a problem with a summary, use the **Give feedback** link next to it.

2.28 GOFlowLLM

GOFlowLLM uses large language models to curate *Gene Ontology annotations* from open access papers that describe miRNA-mRNA interactions. These interactions cover a small part of the GO graph, for which the GO consortium has well defined curation guidelines. GOFlowLLM follows the flowchart in those guidelines step by step, and reproduces the annotations of expert curators in around 90% of cases. The approach is described in GOFlowLLM: curating miRNA literature with large language models and flowcharts.

Browse all sequences with GOFlowLLM annotations →

2.28.1 Annotations on RNAcentral pages

Sequences curated by GOFlowLLM have a **GOFlowLLM** tab. For each paper, it shows the decisions the model made on its way through the flowchart, the evidence it took from the paper, and the model's reasoning. For example, see the results for human hsa-miR-21-5p.

2.28.2 How it works

The flowchart from the GO curation guidelines is encoded as a graph. At each step, the model is asked a yes or no question and given the relevant part of the paper, for example the methods section when looking for an assay. Following the answers leads either to an annotation or to a decision that no annotation can be made. A reasoning model is used, so it can work through the biology before answering.

The model's output is also constrained, so that it can only:

- answer yes or no to each flowchart question;
- quote evidence that appears word for word in the paper;
- choose genes and proteins from those that Europe PMC has tagged in the paper.

When four expert GO curators checked the results for a set of papers, GOFlowLLM reproduced their curation in around 90% of cases. Where it differed from older annotations, this was often because the guidelines had changed since, so GOFlowLLM can also help bring existing annotations up to date.

2.28.3 Limitations

- GOFlowLLM follows the curators' flowchart rigidly, so it is weaker on the parts of the flowchart that curators use least.
- Gene and protein choices depend on the Europe PMC annotations, which are sometimes wrong.
- A paper curated by GOFlowLLM is not fully curated. It only extracts what the miRNA guidelines cover, and a human curator would probably find more. These papers are a good starting set for further curation.

2.29 nf-core/rnastructurome



nf-core/rnastructurome is a bioinformatics pipeline for analysing chemical RNA structure probing data. It takes sequencing reads from SHAPE or DMS experiments, measures how reactive each nucleotide is to the chemical probe, and uses those reactivities to predict RNA secondary structures. It is developed by the RNAcentral team at EMBL-EBI as part of the nf-core community of curated Nextflow pipelines, and its first stable release, 1.0.0, came out in September 2026.

Anyone can run the pipeline on their own data. It is intended as a common standard for processing chemical probing data, so that results from different studies and laboratories are produced in the same way and can be compared directly. RNAcentral uses it to process the datasets described on the *Chemical probing* page.

If you work with an AI agent, the `nfcore-rnastructurome-wrapper` skill in the ClawBio skill library can help you create the samplesheet and the command to run the pipeline. The skill is written around how the pipeline actually behaves, including the samplesheet rules, how the reference is chosen and known gotchas, so the agent can set up a correct run.

2.29.1 What the pipeline does

1. Checks and trims the reads, with optional UMI extraction and deduplication
2. Uses a local reference, or downloads one from Ensembl or NCBI for the sample's organism
3. Aligns the reads to the genome, or to a set of transcripts
4. Counts the reactivity of each nucleotide and normalises it against the untreated or denatured controls
5. Predicts secondary structures from the reactivities and draws 2D diagrams with ViennaRNA, and with *R2DT* where a template exists for the RNA type
6. Exports reactivity, Shannon entropy and base-pair tracks, and RMDB-compatible RDAT files
7. Summarises the run, including how well replicates agree, in a MultiQC report

2.29.2 Usage

The pipeline takes a samplesheet with one row per sample, giving its FASTQ files, sample group, condition (treated, untreated or denatured) and replicate. Many parameters are chosen automatically from what the samplesheet contains, such as the probing method, readout principle and controls present, so the pipeline is suitable for beginners and runs with sensible defaults:

```
nextflow run nf-core/rnastructurome -r 1.0.0 \
   -profile <docker/singularity/.../institute> \
   --input samplesheet.csv \
   --outdir <OUTDIR>
```

For the samplesheet columns, all options and a description of the output files, see the usage, parameter and output documentation.

2.29.3 Credits and support

nf-core/rnastructurome was originally written by Victoria Begley and Pedro Madrigal from RNAcentral at EMBL-EBI, with extensive assistance from Danny Incarnato and Yiliang Ding.

Contributions are welcome: open a pull request on GitHub following the contributing guidelines and the nf-core conventions.

- Source code: https://github.com/nf-core/rnastructurome
- Get help on the `#rnastructurome` Slack channel (join with this invite)

2.30 nf-core/ncrnannotator



Note

nf-core/ncrnannotator is under construction and has not yet had its first release. The documentation linked below tracks the `dev` branch: https://nf-co.re/ncrnannotator/dev/

nf-core/ncrnannotator is a bioinformatics pipeline for annotating non-coding RNA (ncRNA) genes in genome assemblies. It searches a genome with Rfam covariance models using Infernal and reports the ncRNA loci it finds in GTF, GFF3 and BED formats. It is developed by the RNA Resources team at EMBL-EBI as part of the nf-core community of curated Nextflow pipelines.

2.30.1 What the pipeline does

1. Filters the Rfam covariance models to those suited to the chosen annotation mode
2. Splits the genome into chunks so they can be searched in parallel
3. Searches each chunk with Infernal `cmsearch`
4. Removes overlapping hits and applies each Rfam family's gathering (GA) threshold
5. Exports the annotation in GTF, GFF3 and BED formats
6. Summarises the run in a MultiQC report

2.30.2 Annotation modes

The `--mode` parameter sets which Rfam families are searched and how large the genome chunks are:

Mode	Use for	Rfam families
`ensembl-vertebra`	Vertebrate genomes	A curated subset (snRNA, snoRNA, rRNA, SRP RNA, Y RNA, RNase P, vault RNA)
`ensembl-invertel`	Invertebrate genomes	A curated subset, with smaller chunks for compact genomes
`mgnify-assembly`	Metagenomic assemblies	All families, including bacterial and archaeal rRNA
`full`	Any eukaryotic genome	All families, without prokaryotic rRNA

2.30.3 Usage

The pipeline takes a genome FASTA file along with Rfam covariance models and seed alignment files, which can be downloaded from the Rfam FTP site. No samplesheet is needed:

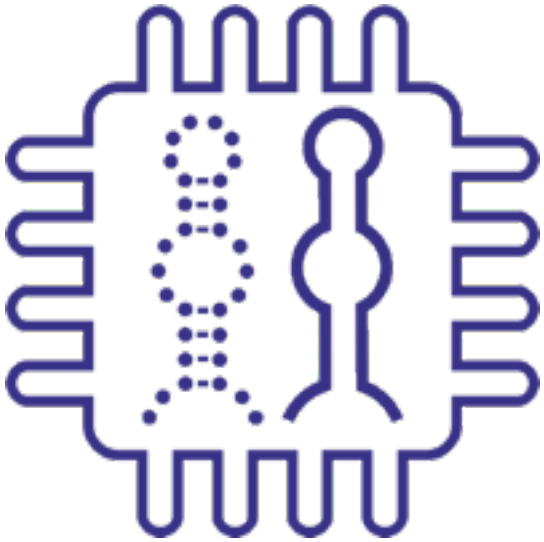
```
nextflow run nf-core/ncrnannotator -r dev \
   -profile <docker/singularity/.../institute> \
   --fasta genome.fa \
   --mode ensembl-vertebrates \
   --rfam_cm Rfam.cm \
   --rfam_seed Rfam.seed \
   --outdir <OUTDIR>
```

For all options and a description of the output files, see the usage, parameter and output documentation.

2.30.4 Credits and support

nf-core/ncrnannotator was written by Pedro Madrigal, Hanh Hoang and Victoria Begley from the RNA Resources team at EMBL-EBI.

- Source code: https://github.com/nf-core/ncrnannotator
- Get help on the `#ncrnannotator` Slack channel (join with this invite)



2.31 R2DT

R2DT draws RNA secondary structure (2D) diagrams in consistent, recognisable layouts, using a library of templates that covers a wide range of RNA types, including rRNA, tRNA, RNase P, snoRNA and the RNA families in Rfam. It was developed by the RNAcentral team at EMBL-EBI, and RNAcentral uses it to draw the diagrams described on the *Secondary structure* page.

For each sequence, R2DT automatically selects a matching template with ribovore, folds the sequence using Infernal and renders the diagram with Traveler.

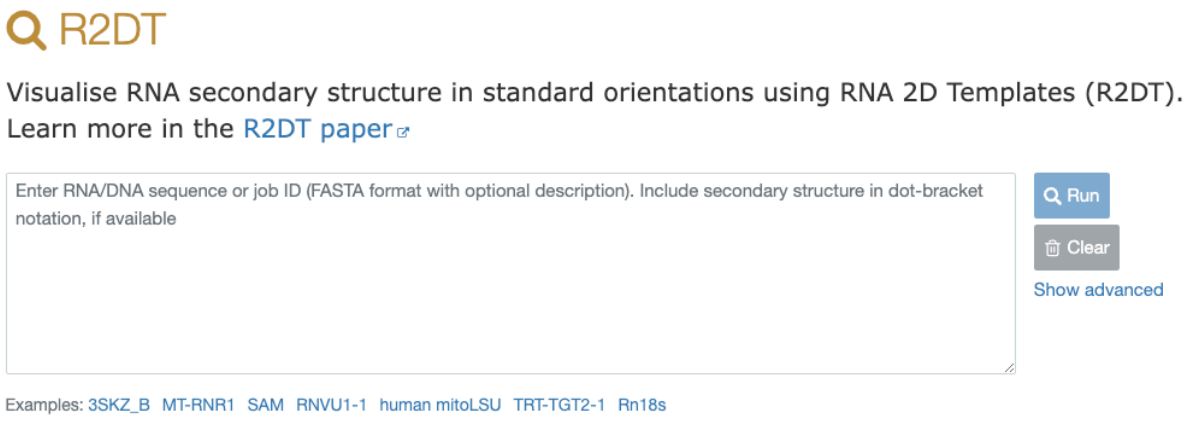
Anyone can use R2DT to visualise the 2D structure of their own sequences:

- in the *R2DT web application* on RNAcentral, or at r2dt.bio;
- through the EMBL-EBI Job Dispatcher API;
- on the command line, with Docker, Apptainer (Singularity) or a local installation.

Beyond drawing single sequences, R2DT can also draw structures from PDB files, process Stockholm alignments, annotate viral genomes and show covariation. For installation, all commands and options, and how to edit the diagrams, see the R2DT documentation.

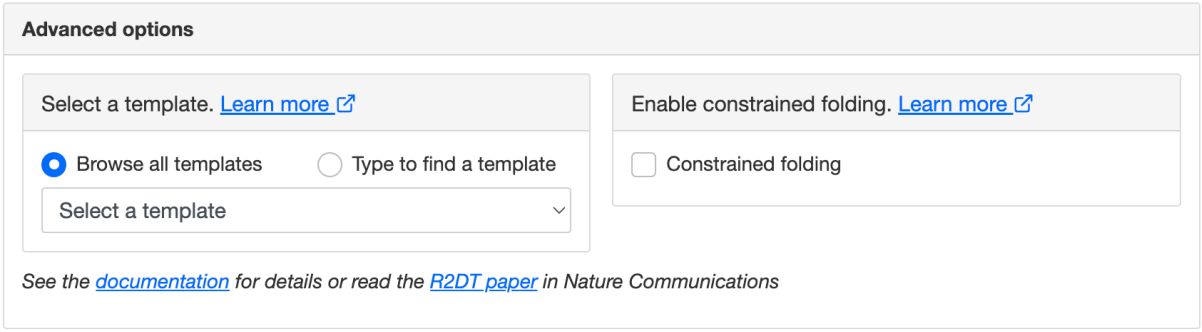
2.31.1 R2DT web application

A secondary structure can be generated for an RNA sequence using the R2DT web application.



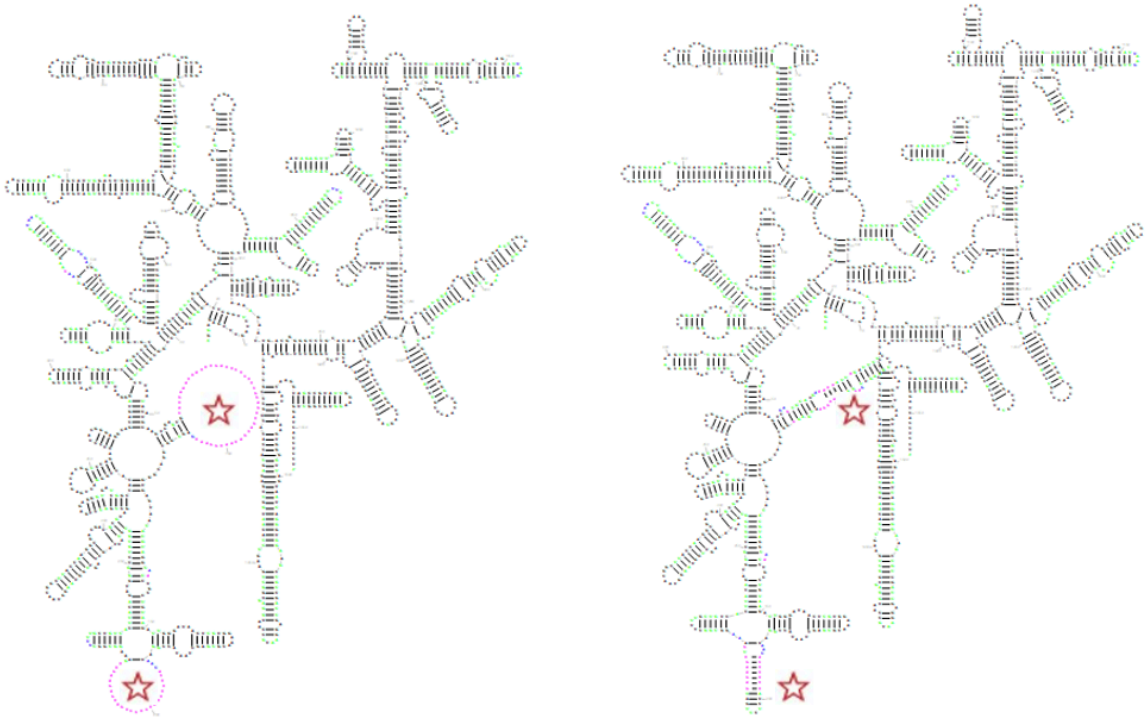
Template selection

By default, R2DT chooses a template automatically, but it is possible to choose a template manually by clicking **Show advanced** and browsing the templates. This option could be useful to highlight species-specific regions. For example, one could display the human small subunit (SSU) rRNA using a bacterial template.



Constrained folding

The advanced options also enable constrained folding, which uses RNAfold from the ViennaRNA package to add base pairs for the single-stranded regions that are not modelled by the templates. For example, this method can improve microRNA diagrams where the Rfam consensus structure has long dangling ends or large hairpin loops. It can also fold species-specific insertions found in many rRNAs:

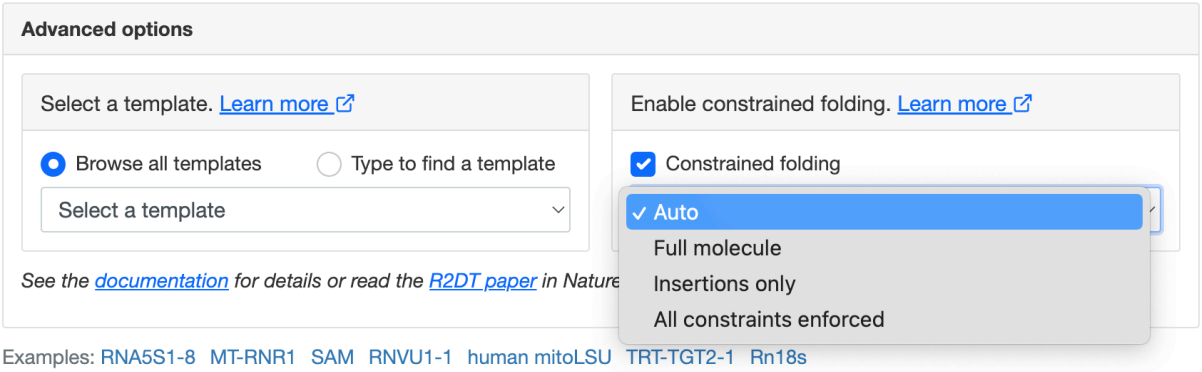


There are four ways of using constrained folding:

- **Auto:** by default, the best mode is chosen automatically based on RNA type and insertion size;
- **Full molecule:** the entire molecule is folded using RNAfold with the template structure provided as a constraint;

- **Insertions only:** the secondary structure of the insertion relative to the template is predicted with RNAfold and added to the diagram;
- **All constraints enforced:** same as above, except that the nucleotides that align to single-stranded regions of the template are kept unpaired when predicting the new structure using RNAfold.

To use constrained folding, choose one of the modes in the advanced options:



2.31.2 Embeddable widget

It is possible to embed an R2DT secondary structure into any website, and it has already been added to FlyBase, SGD and Nucleic Acid Knowledgebase.

Find out more about how to integrate this widget into your website.

2.31.3 Find out more

The method is described in R2DT is a framework for predicting and visualising RNA secondary structure using templates and R2DT: a comprehensive platform for visualizing RNA secondary structure.

- R2DT documentation: https://docs.r2dt.bio
- R2DT web application: https://rnacentral.org/r2dt
- R2DT source code: https://github.com/r2dt-bio/R2DT

2.32 Chemical probing

Chemical probing experiments, such as SHAPE (Selective 2'-Hydroxyl Acylation Analyzed by Primer Extension) and DMS (Dimethyl Sulfate), measure how reactive each nucleotide of an RNA is to a chemical probe, which shows which parts of the RNA are paired and which are flexible. RNAcentral is bringing these data together: the results are planned to appear in RNAcentral in mid 2027 (release 31), in a new chemical probing tab on sequence pages and as reactivity tracks in the *genome browser*. This will make RNAcentral a central hub for chemical probing data.

2.32.1 A metadata standard for probing experiments

Chemical probing datasets are often deposited with incomplete or inconsistent descriptions of how they were produced, which makes them hard to find, reuse and compare. To address this, the rnacentral-probing-metadata repository sets out a **standard for describing probing experiments**. Every dataset is described by a metadata file that follows a shared schema, recording:

- the organism (and, for viruses, the strain);
- the probing method and chemical, the readout principle and the reverse transcriptase;

- a term from the Ontology for Biomedical Investigations (OBI) for the assay, such as OBI:0003886 (NAI RNA structure mapping assay), so the type of experiment is recorded in a standard vocabulary shared with other resources;
- the experimental context, such as *in vivo* or *in vitro*, and the RNA fraction sequenced;
- the publication;
- the raw data accessions, with each sample's group, condition and replicate.

More than 140 SHAPE and DMS datasets are already described this way.

Anyone can add their own dataset by opening a pull request with a new metadata file, following the instructions and template in the repository. Each file is checked automatically against the schema and then reviewed by the RNAcentral team. The raw data must be in a public archive such as ENA, SRA or GEO, and each treated sample group needs at least two biological replicates.

2.32.2 Processing the data

Once a dataset is accepted, its raw data are downloaded and processed with *nf-core/rnastructurome*, a pipeline developed by the RNAcentral team, against reference sequences and annotations built from RNAcentral, which are kept in the rnacentral-probing-references repository. Where possible, the predicted structures are compared with known structures from Rfam to assess how well the probing data support them.

2.33 Linking to RNAcentral

RNAcentral can be linked to from other websites and tools, using stable URLs for sequences and genome locations, embeddable widgets, and RNAcentral icons and logos.

2.33.1 Linking to sequences

Link by external accession

It is possible to link to RNAcentral **without knowing the RNAcentral accession** using a URL in the following format:

```
https://rnacentral.org/link/<database>:<accession>
```

where `database` is one of the RNAcentral *Expert Databases*, and `accession` is an identifier from that database (see examples below). Both parameters are **case-insensitive**.

This functionality relies on the RNAcentral *text search*. If only 1 sequence matches the database and accession, then the link will redirect to that sequence. If several sequences match, then the link will show a search result listing all matching sequences.

Examples

- /link/mirbase:hsa-mir-7161-3p opens the matching sequence
- /link/ensembl:ensg00000202354.1 opens the matching sequence
- /link/flybase:FBtr0304468 lists the two matching sequences

Direct link format

Please use the following format to link to RNAcentral sequences:

```
https://rnacentral.org/rna/<RNAcentral identifier>/<NCBI taxonomy␣
↪identifier>
```

The URL includes NCBI taxonomy identifiers, for example *9606* for *Homo sapiens*.

Either slash / or underscore _ can be used as a separator:

- URS000075A546/9606
- URS000075A546_9606

Links without NCBI taxonomy identifiers show annotations from all species where the sequence was found.

You can optionally add the RNAcentral logo before the link, for example:

Example code:

```
<p>
  <img src="https://rnacentral.org/static/img/logo/rnacentral-logo.png"␣
↪style="width: 16px; height: 16px; vertical-align: middle;">
  <a href="https://rnacentral.org/rna/URS000075A546/9606">URS000075A546/9606
↪</a>
</p>
```

RNAcentral logos are available in the *Icons and logos* section.

Mapping between third-party identifiers and RNAcentral IDs

For a **small number of sequences**, you can find RNAcentral accessions using *text search*.

If you need to perform a **large-scale mapping** between RNAcentral identifiers and Expert Database accessions, please use the mapping files found in the *FTP Archive*:

- Mapping between RNAcentral sequences and all Expert Databases in a **single file**: https://ftp.ebi.ac.uk/pub/databases/RNAcentral/current_release/id_mapping/
- Mapping between RNAcentral sequences and each Expert Database in **separate files**: https://ftp.ebi.ac.uk/pub/databases/RNAcentral/current_release/id_mapping/database_mappings/

2.33.2 Linking to genome locations

You can link to a genome location of interest within the RNAcentral *genome browser* using links like this:

```
https://rnacentral.org/genome-browser?species=<species>&chromosome=
↪<chromosome>&start=<start>&end=<end>
```

where:

- `<species>` is the organism name in lowercase, with spaces replaced by underscores as in Ensembl, for example *homo_sapiens* or *canis_lupus_familiaris*;
- `<chromosome>` is the chromosome name in Ensembl format, for example `1` (not `chr1`);
- `<start>` and `<end>` are the genome coordinates.

Example:

Example code:

```
<p>
  <img src="https://rnacentral.org/static/img/logo/rnacentral-logo.png"␣
↪style="width: 16px; height: 16px; vertical-align: middle;">
  <a href="https://rnacentral.org/genome-browser?species=homo_sapiens&
↪chromosome=X&start=73819307&end=73856333">Human XIST gene</a>
</p>
```

2.33.3 Embeddable widgets

Several RNAcentral tools are available as web components that can be added to any website. The instructions for each are on GitHub:

Widget	What it shows	Instructions
Sequence search	The *sequence similarity search*, optionally restricted to your own database	rnacentral-sequence-search-embed
R2DT	*Secondary structure* diagrams for a sequence	r2dt-web
LitScan	The *articles* that mention a list of identifiers	rnacentral-litscan

2.33.4 Icons and logos

RNAcentral Expert Database badge

Databases *integrated in RNAcentral* can display an **RNAcentral Expert Database badge**:

Download the badge (svg)

Example code:

```
<a href="https://rnacentral.org">
  <img src="https://rnacentral.org/static/img/logo/rnacentral-expert-
↪database.svg" style="width: 140px">
</a>
```

The badge should link to the RNAcentral homepage or to the **landing page** of the member database within RNAcentral, such as https://rnacentral.org/expert-database/flybase or https://rnacentral.org/expert-database/gtrnadb.

RNAcentral icons

The RNAcentral symbol in different resolutions:

- 16x16
- 24x24
- 32x32
- 64x64
- 128x128
- 530x530

RNAcentral logos

The full logo comes in three versions, which differ only in the colour of the word “central”:

- White, for dark backgrounds
- Dark grey, for light backgrounds
- Light grey, a smaller version (600 × 100 pixels) for light backgrounds

2.34 Licence

RNAcentral wants to ensure that the information is accessible to the public and free to use. For this reason, from **release 20 onwards**, all data have been released under the Creative Commons Zero licence (CC0). This means that **any form of reuse of the content is permitted**. Data from releases before 20 remain under EMBL-EBI’s Terms of use.

For more information, see the CC0 FAQ.

2.34.1 Long-term data preservation

RNAcentral follows the EMBL-EBI long-term data preservation policy.

2.34.2 Attribution

Attribution isn’t required, but **we recommend citing RNAcentral** when you use its services (such as sequence search), data or software. See *Publications* for the recommended citation and the full list of RNAcentral papers.

2.35 Browser compatibility

RNAcentral is designed to work best in the following browsers:

- Chrome
- Firefox
- Safari
- Microsoft Edge

Keeping your browser up to date gives the best experience on the RNAcentral website. Most browsers update themselves automatically; check your browser’s About or Help menu to confirm the version you are running.

2.36 Contact us

Please feel free to send your feedback about any aspect of RNAcentral. We really value your input and will get back to you as soon as we can.

Get in touch →

Every page on the RNAcentral website also has a **Feedback** button in the top right corner.

2.36.1 Report a website problem

If you would like to report a problem with the website, please use the **Feedback** button or the contact form.

2.36.2 Documentation feedback

If something on these pages is unclear, missing or incorrect, let us know through the contact form above.



2.36.3 RNAcentral blog

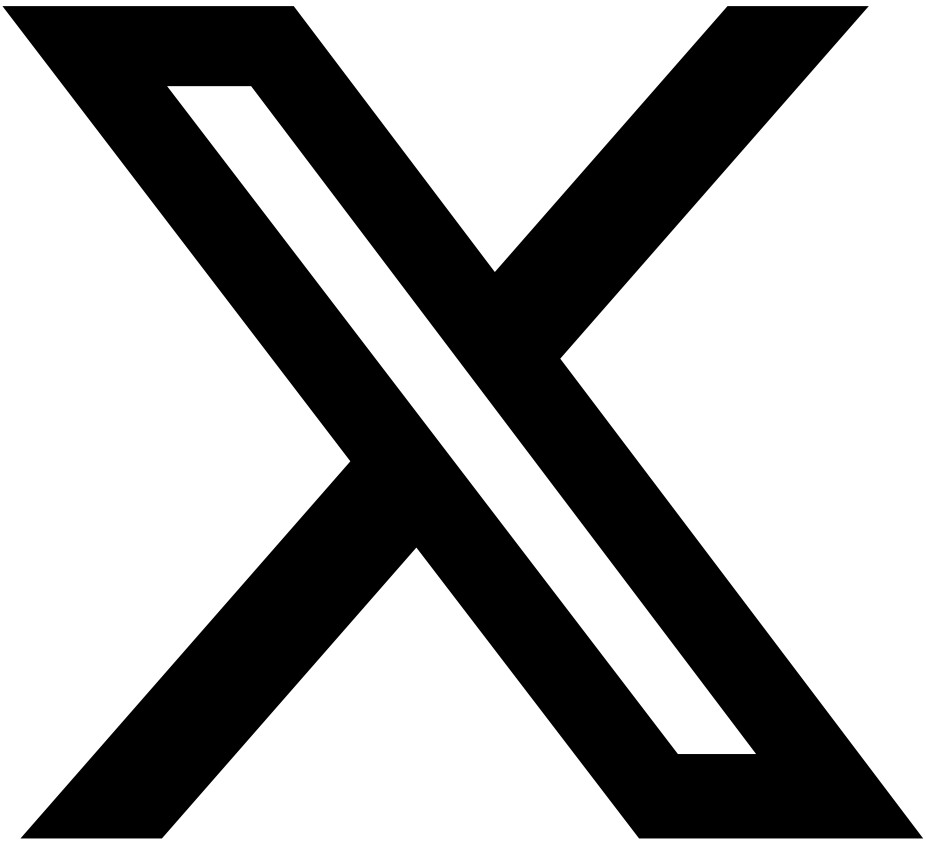
New releases, features and other news are announced on the RNAcentral blog.

2.36.4 Social media

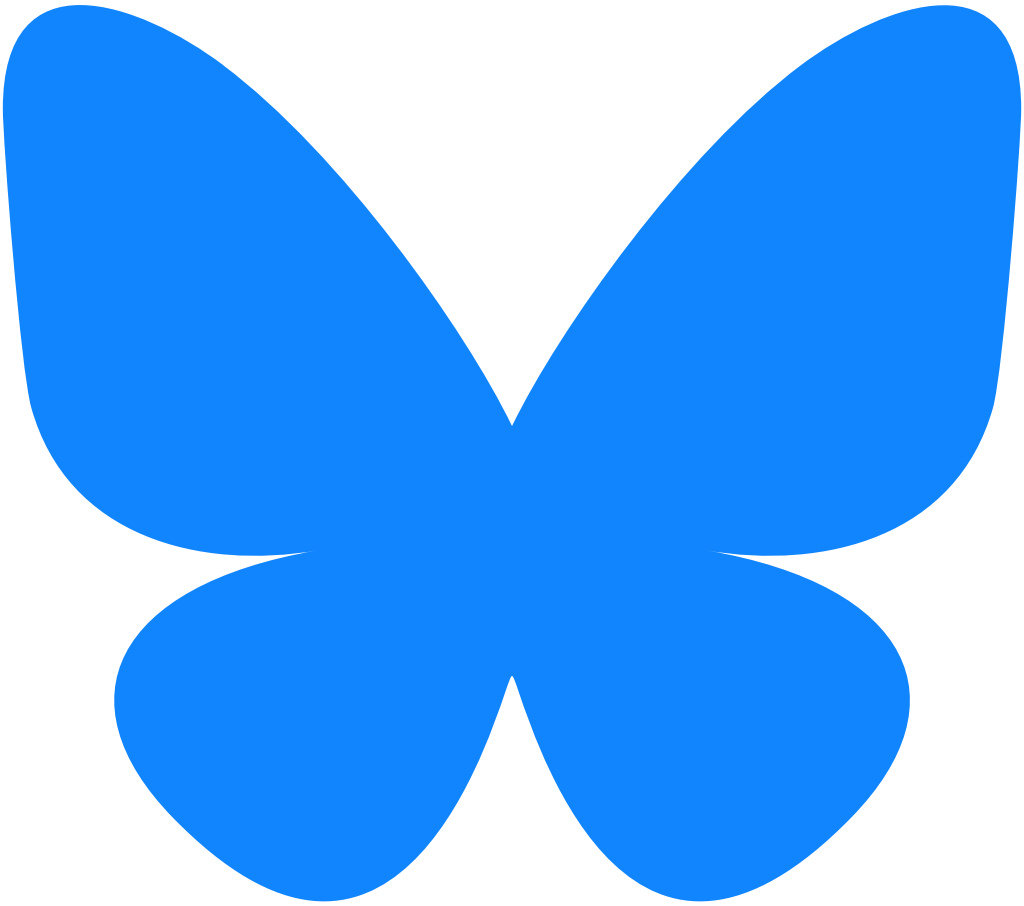
Follow us for updates on any of these channels:



- **LinkedIn:** EMBL-EBI RNA Resources



- **X** (formerly Twitter): @rnacentral



- **Bluesky:** @rnacentral.bsky.social