

GAMA: Genomic Availability & Metadata Analysis Tool (v1)

Overview

The Genomic Availability & Metadata Analysis Tool (GAMA) is an R-based framework for efficiently surveying publicly accessible sequencing data for user-defined sets of species. It unifies NCBI Assembly, SRA, and BioSample query searches to generate reproducible availability summaries with a composite ‘data richness’ score, alongside ontology-based breakdowns of SRA modalities and GEO experiment linkage. GAMA is intended to support feasibility assessments of *in silico* research on underutilised plants.

Core Functionality

GAMA (v1) incorporates two modules:

Availability Query Tool – Interrogates NCBI databases using `rentrez` to quantify per-species data availability. Searches use organism-restricted terms with `retmax = 999999` and history tracking enabled. All queries are wrapped in safe handlers. Accession summaries are retrieved in batches of 100 using `entrez_summary()` with up to ten retries and exponential backoff. Query provenance (tool version, timestamp, databases, and terms) is attached to all outputs.

`query_species()`

- Input – character vector of binomial species names
- Output – list object of search results, accession counts, record IDs, and query metadata

`summarise_availability()`

- Input – list object returned by `query_species()`
- Output – tibble (`gdt_tbl`) containing accession counts and scores

`plot_availability()`

- Input – tibble returned by `summarise_availability()`
 - Optional arguments –
 - `rank`: character; species ordering method
 - `abbreviate`: logical; abbreviate species names
 - `theme_fn`: ggplot2 theme function
 - `colours`: named character vector of fill colours
- Output – ggplot object displaying stacked bar charts of data richness

Metadata Analysis Tool – Retrieves SRA metadata by parsing the `expxml` field using `xml2`, extracting and normalising `LIBRARY_STRATEGY` values, and assigning ontology-based modality class and subclass using strict matching. When `LIBRARY_STRATEGY` is missing, uninformative, or explicitly ‘other’, a fallback rescue is applied using `LIBRARY_SOURCE`, `LIBRARY_SELECTION`, and `TITLE` fields. Experiments remain unknown when no reliable signal is present, while assays that do not match any ontological category are retained as other. GEO linkage is recorded by scanning experiment XML for GSE/GSM accessions without retrieving full GEO records. Query provenance is attached to all outputs.

`extract_assembly_metadata()`

- Input – list object returned by `query_species()`
 - Optional arguments –
 - `species`: character vector of species to include
 - `best`: logical; return only the best assembly per species
- Output – tibble containing assembly-level metadata (species, accession, level, N50, coverage, BioSample, BioProject, submitter, release date, FTP path)

`extract_sra_metadata()`

- Input – list object returned by `query_species()`
 - Optional arguments –
 - `species`: character vector of species to include
 - `class`: character vector of ontology classes to retain
 - `subclass`: character vector of ontology subclasses to retain
 - `only_geo`: logical; retain only GEO-linked experiments
- Output – tibble containing experiment-level SRA metadata with raw/normalised strategies, ontology class/subclass assignments, and GEO linkage fields

`summarise_sra_availability()`

- Input – list object returned by `query_species()`
 - Optional arguments –
 - `species`: character vector of species to include
 - `all`: logical; include subclass-level counts
 - `include_geo`: logical; append GEO linkage summaries
- Output – tibble containing species-level SRA modality counts and total SRA; optionally includes subclass and GEO overlay columns

`plot_sra_availability()`

- Input – tibble returned by `summarise_sra_availability()`
 - Optional arguments –
 - `species`: character vector of species to include
 - `rank`: character; species ordering method
 - `abbreviate`: logical; abbreviate species names
 - `theme_fn`: ggplot2 theme function
 - `colours`: named character vector of class colours
- Output – ggplot object showing proportional SRA modality composition across species

`plot_sra_geo_availability()`

- Input – tibble returned by `summarise_sra_availability(include_geo = TRUE)`
 - Optional arguments –
 - `species`: character vector of species to plot
 - `classes`: character vector of modality classes to display
 - `rank`: character; species ordering method
 - `theme_fn`: ggplot2 theme function
 - `colours`: named character vector of class colours
 - `alpha_vals`: named numeric vector controlling GEO-linked transparency
- Output – ggplot object (single species) or named list of ggplot objects showing per-modality GEO linkage

Data Richness

$Score = A + S + B$, where A , S , and B are the transformed contributions of Assembly, SRA, and BioSample accession counts. $A = best + \ln(1 + total - best)$, with assemblies weighted as Complete = 10, Chromosome = 8, Scaffold = 5, and Contig = 2; $best$ is the maximum weight assembly, with ties broken by highest N50, and $total$ is the sum of all accession weights. $S = 2 \cdot \ln(1 + SRA)$, and $B = \ln(1 + BioSample)$. This formulation prioritises high-quality assemblies while incorporating diminishing returns for highly sampled taxa.

Ontology

Subclasses and recognised term variants for experimental modality were derived by mining >220,000 *Arabidopsis thaliana* SRA accessions (last updated: 31 January 2026) before manual curation to capture common submitter variants and deprecated terminology (Tab. 1).

Table 1. Ontology for assigning SRA experiments to modality classes and subclasses based on normalised LIBRARY_STRATEGY terms.

Class	Subclass	Recognised Library Strategy Terms (normalised variants)
Genomic	WGS	wgs, wga, wcs, wxs, finishing; whole genome sequencing/seq/shotgun/resequencing; genome/genomic sequencing
	Amplicon-seq	amplicon; amplicon seq/sequencing; targeted amplicon; 16S/18S/ITS amplicon/sequencing; metabarcoding
	RAD-seq	rad seq/sequencing/radseq; restriction site associated sequencing; ddRAD/ddRADseq; GBS; genotyping by sequencing/seq
	Targeted-Capture	targeted capture/sequencing/resequencing; exome seq/sequencing/WES; capture seq/sequencing; hybrid capture; target enrichment; panel sequencing; gene panel
	Clone-based	clone/cloneend/poolclone; clone end(s); BAC/fosmid/cosmid end seq; pool clone sequencing
Transcriptomic	RNA-seq	rna seq/rnaseq/sequencing; transcriptome seq/sequencing/profiling; total/stranded/polyA RNA-seq; cDNA seq; full-length cDNA; scRNA/snRNA; tag seq; DGE
	small-RNA	miRNA/siRNA/piRNA; small RNA seq/sequencing; ncRNA; lncRNA; smRNA
	Long-read	Iso-Seq/isodeq; direct RNA sequencing; nanopore cDNA; full-length transcriptome
Epigenomic	Bisulfite-seq	bisulfite seq/sequencing; WGBS/RRBS; methylation/methylome seq; MBD/MeDIP/MRE seq
	ChIP-seq	chip seq/chipseq/sequencing; ChIP-exo; RIP-seq; RNA immunoprecipitation seq
	CUT&RUN	cut run; cutandrun; cut run sequencing
	CUT&Tag	cut tag; cutandtag; cut tag sequencing
	ATAC-seq	atac seq/atacseq/sequencing; assay for transposase accessible chromatin; scATAC/snATAC
	DNase-seq	dnase hypersensitivity; dnase I seq/sequencing
	FAIRE-seq	faire seq/sequencing; formaldehyde-assisted isolation
	MNase-seq	mnase seq/sequencing; micrococcal nuclease seq; nucleosome mapping
	SELEX	selex; ht selex; selex seq/sequencing
Chromatin	Hi-C	hi-c/hic; hi-c seq/sequencing; chromosome conformation capture carbon copy
	3C-based	3C/4C/5C; capture C; promoter capture C; HiChIP; PLAC-seq
	ChIA-PET	chia-pet; chia-pet seq/sequencing; chromatin interaction analysis by paired-end tag
	TCC	tethered chromatin conformation capture; tcc seq/sequencing
Other	Other	custom protocols; metagenomic/metatranscriptomic; proteomic/metabolomic; arrays; imaging; optical mapping; PCR/qPCR/RT-PCR; library construction; validation; test/pilot/control; spike-in

Software Requirements

R packages:

- dplyr – data manipulation
- ggplot2 – plotting
- purrr – functional iteration
- rentrez – NCBI querying
- tibble – tidy table creation
- tidyr – pivoting and reshaping
- xml2 – XML parsing and tag extraction

R \geq 4.1 is recommended.

Quick-Start Example

1. Download and run both modules.
2. Query NCBI databases using a list of species.

```
RESULTS <- query_species(c(  
  'Phaseolus vulgaris',  
  'Vigna radiata',  
  'Vigna unguiculata',  
  'Vigna angularis',  
  'Vigna vexillata'  
))
```

3. Summarise data richness.

```
SUMMARY <- summarise_availability(RESULTS)  
SUMMARY
```

4. Visualise data richness.

```
plot_availability(SUMMARY)
```

5. Summarise SRA modality composition.

```
META <- summarise_sra_availability(RESULTS)  
META
```

6. Visualise SRA modality composition.

```
plot_sra_availability(META)
```

7. Extract filtered Assembly accession metadata.

```
extract_assembly_metadata(RESULTS, best = TRUE)
```

8. Extract filtered SRA accession metadata.

```
extract_sra_metadata(RESULTS, species = 'Vigna vexillata', class = 'genomic')
```

Note: Large or high-profile species sets may require extended runtimes.

References

Müller K., Wickham H. (2026) tibble: Simple Data Frames. R package version 3.3.1.

National Center for Biotechnology Information (2026) Entrez Programming Utilities. National Library of Medicine. Available at: <https://www.ncbi.nlm.nih.gov> (Accessed: 31 January 2026).

R Core Team (2024) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Wickham H. (2016) ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag, New York.

Wickham H., François R., Henry L., Müller K., Vaughan D. (2023) dplyr: A Grammar of Data Manipulation. R package version 1.1.4.

Wickham H., Henry L. (2026) purrr: Functional Programming Tools. R package version 1.2.1.

Wickham H., Hester J., Ooms J. (2026) xml2: Parse XML. R package version 1.5.2.

Wickham H., Vaughan D., Girlich M. (2025) tidyr: Tidy Messy Data. R package version 1.3.2.

Winter D.J. (2017) rentrez: An R package for the NCBI eUtils API. *The R Journal*, 9: 520-526.