

# Package: AIGRA (via r-universe)

June 5, 2026

**Title** Agentic Item Generation, Review, and Analysis

**Version** 0.1.2

**Author** Moses O. Omopekunola [aut, cre]

**Maintainer** Moses O. Omopekunola <omopekunola.m@hse.ru>

**Description** Provides tools for validating, generating, reviewing, reporting, and visualising assessment item generation workflows. The package supports tabular item-bank templates, item-bank validation, 'Python'-backed agentic generation workflows, multimodal diagram generation, quality summaries, and 'HTML' reporting. External artificial intelligence services and related 'API' calls require user-supplied credentials and are not called during package checks. The workflow is informed by automatic item generation methods described by Gierl and Haladyna (2013, ISBN:9780415897518) and evidence-centered assessment design described by Mislevy et al. (2003) <doi:10.1002/j.2333-8504.2003.tb01908.x>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** graphics, grid, reticulate, utils, grDevices,

**Suggests** base64enc, png, readxl, rmarkdown, writexl

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-05 11:40:24 UTC

**RemoteUrl** <https://github.com/cran/AIGRA>

**RemoteRef** HEAD

**RemoteSha** f91304d5878c8a16808b0c597b68ff4973ccec24

## Contents

aigra_apply_diagram_agent . . . . .	3
-------------------------------------	---

<code>aigra_backend_help</code> . . . . .	3
<code>aigra_backend_path</code> . . . . .	4
<code>aigra_build_diagram_prompt</code> . . . . .	4
<code>aigra_build_diagram_prompt_from_row</code> . . . . .	5
<code>aigra_create_sample_source_diagrams</code> . . . . .	5
<code>aigra_detect_diagram_required</code> . . . . .	6
<code>aigra_example_item_bank</code> . . . . .	6
<code>aigra_find_uv</code> . . . . .	7
<code>aigra_generate_diagram</code> . . . . .	7
<code>aigra_generate_diagram_fallback</code> . . . . .	8
<code>aigra_generate_items</code> . . . . .	8
<code>aigra_generate_multimodal_tabular_items</code> . . . . .	10
<code>aigra_generate_result_diagrams</code> . . . . .	11
<code>aigra_generate_result_diagrams_auto</code> . . . . .	12
<code>aigra_generate_tabular_items</code> . . . . .	13
<code>aigra_image_models</code> . . . . .	14
<code>aigra_latest_csv</code> . . . . .	14
<code>aigra_latest_output</code> . . . . .	15
<code>aigra_list_outputs</code> . . . . .	15
<code>aigra_localize_diagram_prompts</code> . . . . .	15
<code>aigra_multimodal_template_items</code> . . . . .	16
<code>aigra_output_dir</code> . . . . .	16
<code>aigra_outputs</code> . . . . .	17
<code>aigra_parse_items</code> . . . . .	17
<code>aigra_parse_tabular_items</code> . . . . .	18
<code>aigra_plot_diagram</code> . . . . .	18
<code>aigra_plot_summary</code> . . . . .	19
<code>aigra_print_summary</code> . . . . .	19
<code>aigra_print_validation</code> . . . . .	20
<code>aigra_python_info</code> . . . . .	20
<code>aigra_read_latest_csv</code> . . . . .	21
<code>aigra_read_latest_output</code> . . . . .	21
<code>aigra_repair_diagram_prompts</code> . . . . .	21
<code>aigra_set_api_keys</code> . . . . .	22
<code>aigra_set_backend</code> . . . . .	23
<code>aigra_show_result_diagram</code> . . . . .	23
<code>aigra_status</code> . . . . .	24
<code>aigra_summarise_output</code> . . . . .	24
<code>aigra_template_items</code> . . . . .	25
<code>aigra_translate_diagram_prompts</code> . . . . .	25
<code>aigra_use_backend</code> . . . . .	26
<code>aigra_validate_tabular_items</code> . . . . .	26
<code>aigra_write_admin_html</code> . . . . .	27
<code>aigra_write_multimodal_template_excel</code> . . . . .	27
<code>aigra_write_report</code> . . . . .	28
<code>aigra_write_template_csv</code> . . . . .	29
<code>aigra_write_template_excel</code> . . . . .	29
<code>ensure_aigra_python</code> . . . . .	30

---

`aigra_apply_diagram_agent`

*Apply AIGRA Diagram Agent to generated results*

---

**Description**

Inspects generated items and automatically sets `diagram_required` and `diagram_prompt` when a visual diagram is likely needed.

**Usage**

```
aigra_apply_diagram_agent(result, overwrite_prompt = FALSE)
```

**Arguments**

`result`            A data frame returned by AIGRA.  
`overwrite_prompt`    If TRUE, replace existing diagram prompts.

**Value**

Updated result data frame.

---

`aigra_backend_help`

*Show 'AIGRA' Backend Setup Help*

---

**Description**

Prints a short guide for configuring the external 'AIGRA' backend.

**Usage**

```
aigra_backend_help()
```

**Value**

Invisibly returns NULL.

**Examples**

```
aigra_backend_help()
```

aigra\_backend\_path     *Get AIGRA backend path*

---

**Description**

Returns the configured AIGRA Python backend path.

**Usage**

```
aigra_backend_path()
```

**Details**

AIGRA first checks the active session configuration. If no backend has been set in the current R session, it checks the AIGRA\_BACKEND\_PATH environment variable.

**Value**

Backend path as a character string.

---

aigra\_build\_diagram\_prompt  
*Build a safe diagram prompt from an item stem*

---

**Description**

Build a safe diagram prompt from an item stem

**Usage**

```
aigra_build_diagram_prompt(  
  stem,  
  option_A = "",  
  option_B = "",  
  option_C = "",  
  option_D = ""  
)
```

**Arguments**

stem	Item stem.
option_A	Option A.
option_B	Option B.
option_C	Option C.
option_D	Option D.

**Value**

A diagram-generation prompt.

---

aigra\_build\_diagram\_prompt\_from\_row  
*Build a diagram prompt from a result row*

---

**Description**

Build a diagram prompt from a result row

**Usage**

```
aigra_build_diagram_prompt_from_row(result, row = 1)
```

**Arguments**

result	A data frame returned by AIGRA.
row	Row number.

**Value**

A diagram prompt.

---

aigra\_create\_sample\_source\_diagrams  
*Create source diagrams for the AIGRA multimodal template*

---

**Description**

Create source diagrams for the AIGRA multimodal template

**Usage**

```
aigra_create_sample_source_diagrams(diagram_dir)
```

**Arguments**

diagram_dir	Directory where source diagram PNG files should be saved.
-------------	---

**Value**

Named character vector of diagram paths.

aigra\_detect\_diagram\_required

*Detect whether an item likely requires a diagram*

---

### **Description**

Uses rule-based visual triggers in the item stem, topic, objective, and options.

### **Usage**

```
aigra_detect_diagram_required(stem, topic = "", objective = "", options = "")
```

### **Arguments**

stem	Item stem.
topic	Optional topic.
objective	Optional objective.
options	Optional option text.

### **Value**

TRUE or FALSE.

---

aigra\_example\_item\_bank

*Get path to bundled AIGRA example item bank*

---

### **Description**

Get path to bundled AIGRA example item bank

### **Usage**

```
aigra_example_item_bank()
```

### **Value**

Path to the bundled example CSV item bank.

---

aigra_find_uv	<i>Find uv executable</i>
---------------	---------------------------

---

**Description**

Find uv executable

**Usage**

```
aigra_find_uv()
```

**Value**

Path to uv executable.

---

aigra_generate_diagram	<i>Generate a single AIGRA diagram from a prompt</i>
------------------------	--

---

**Description**

Generate a single AIGRA diagram from a prompt

**Usage**

```
aigra_generate_diagram(  
    prompt,  
    output_path,  
    provider = "gemini",  
    model = "gemini-2.5-flash-image",  
    size = "1024x1024"  
)
```

**Arguments**

prompt	Diagram prompt.
output_path	Output PNG path.
provider	Image provider.
model	Image model.
size	Image size.

**Value**

Output path.

aigra\_generate\_diagram\_fallback

*Generate a diagram with fallback image models*

---

### Description

Tries multiple image-generation models in order until one succeeds.

### Usage

```
aigra_generate_diagram_fallback(  
  prompt,  
  output_path,  
  provider = "gemini",  
  models = c("gemini-2.5-flash-image", "gemini-3.1-flash-image-preview",  
             "gemini-3-pro-image-preview"),  
  size = "1024x1024"  
)
```

### Arguments

prompt	Diagram prompt.
output_path	Output PNG path.
provider	Image provider.
models	Character vector of model names to try in order.
size	Image size.

### Value

Output path.

---

aigra\_generate\_items *Generate Assessment Item Clones with Simplified API-Key Handling*

---

### Description

A user-friendly wrapper around `aigra_generate_tabular_items()` that allows users to provide provider API keys directly in R.

**Usage**

```
aigra_generate_items(
  file_path,
  model = "gemini-3.1-pro-preview",
  provider = "auto",
  gemini.API = NULL,
  openai.API = NULL,
  groq.API = NULL,
  anthropic.API = NULL,
  backend_path = NULL,
  ...
)
```

**Arguments**

<code>file_path</code>	Path to the item-bank template file.
<code>model</code>	Model name or alias. For example, "gemini", "sonnet", "gpt4o", or "llama".
<code>provider</code>	Provider name. Use "auto" to infer the provider from the model name or supplied API key.
<code>gemini.API</code>	Optional 'Gemini' API key.
<code>openai.API</code>	Optional 'OpenAI' API key.
<code>groq.API</code>	Optional 'Groq' API key.
<code>anthropic.API</code>	Optional 'Anthropic' API key.
<code>backend_path</code>	Optional path to the external 'AIGRA_BACKEND' folder.
<code>...</code>	Additional arguments passed to <code>aigra_generate_tabular_items()</code> , such as <code>source_language</code> , <code>target_language</code> , <code>subject</code> , <code>exam</code> , <code>n_clones</code> , and <code>max_items</code> .

**Value**

A data frame of generated items and review information.

**Examples**

```
if (interactive()) {
  out <- aigra_generate_items(
    file_path = "items.xlsx",
    model = "sonnet",
    anthropic.API = "your_key",
    backend_path = "path/to/AIGRA_BACKEND",
    source_language = "English",
    target_language = "English",
    subject = "Mathematics",
    exam = "Demo",
    n_clones = 1,
    max_items = 2
  )
}
```

---

```
aigra_generate_multimodal_tabular_items
```

*Generate multimodal items from a tabular item bank*

---

## Description

Runs item generation and diagram generation in one workflow.

## Usage

```
aigra_generate_multimodal_tabular_items(
  file_path,
  provider = "gemini",
  model = "gemini-3.1-pro-preview",
  image_provider = "gemini",
  image_model = "gemini-3-pro-image-preview",
  source_language = "English",
  target_language = "English",
  subject = "General",
  exam = "AIGRA Multimodal Item Bank",
  n_clones = 1,
  max_items = NULL,
  max_images = NULL,
  write_reports = TRUE,
  include_key = TRUE,
  only_accepted = TRUE
)
```

## Arguments

<code>file_path</code>	Path to CSV or Excel item bank.
<code>provider</code>	Text-generation provider.
<code>model</code>	Text-generation model.
<code>image_provider</code>	Image-generation provider.
<code>image_model</code>	Image-generation model.
<code>source_language</code>	Source item language.
<code>target_language</code>	Target generated item language.
<code>subject</code>	Subject name.
<code>exam</code>	Exam or item-bank name.
<code>n_clones</code>	Number of clones per item.
<code>max_items</code>	Maximum source items to process. Use NULL for all.
<code>max_images</code>	Maximum diagrams to generate. Use NULL for all generated rows.

write\_reports If TRUE, writes quality and administration HTML reports.  
 include\_key If TRUE, include answer key in administration HTML.  
 only\_accepted If TRUE, administration HTML includes only ok/edited rows.

### Details

Text-only items are generated normally. Items that require diagrams are processed through the Source Diagram Agent and Clone Diagram Agent.

### Value

A list containing result, result\_with\_diagrams, report\_path, admin\_file, and student\_file.

---

aigra\_generate\_result\_diagrams  
*Generate diagrams for rows in an AIGRA result*

---

### Description

Generates diagram images for result rows and returns the updated data frame with diagram\_path, diagram\_prompt, image\_provider, and image\_model columns.

### Usage

```

aigra_generate_result_diagrams(
  result,
  output_dir = NULL,
  provider = "gemini",
  model = "gemini-2.5-flash-image",
  rows = NULL,
  only_required = TRUE,
  force = FALSE,
  max_images = 1,
  overwrite = FALSE
)
  
```

### Arguments

result	A data frame returned by AIGRA.
output_dir	Directory for generated images.
provider	Image provider.
model	Image model.
rows	Optional row numbers to process.
only_required	If TRUE, process only rows where diagram_required is true.
force	If TRUE, generate even when diagram_required is false or missing.
max_images	Maximum number of images to generate.
overwrite	If TRUE, overwrite existing image files.

**Value**

Updated result data frame.

---

aigra\_generate\_result\_diagrams\_auto

*Generate diagrams using the AIGRA Diagram Agent*

---

**Description**

Applies the Diagram Agent, then generates diagrams for rows marked as requiring diagrams.

**Usage**

```
aigra_generate_result_diagrams_auto(  
  result,  
  provider = "gemini",  
  model = "gemini-3-pro-image-preview",  
  rows = NULL,  
  max_images = 3,  
  overwrite = FALSE  
)
```

**Arguments**

result	A data frame returned by AIGRA.
provider	Image provider.
model	Image model.
rows	Optional row numbers to process.
max_images	Maximum number of images to generate.
overwrite	If TRUE, overwrite existing image files.

**Value**

Updated result data frame with diagram paths.

---

`aigra_generate_tabular_items`*Generate items from a tabular item bank*

---

## Description

Runs the AIGRA generation, solver, critic, and export pipeline using a CSV or Excel item bank.

## Usage

```
aigra_generate_tabular_items(  
    file_path,  
    provider = "gemini",  
    model = "gemini-3.1-pro-preview",  
    source_language = "English",  
    target_language = "English",  
    review_language = "English",  
    subject = "General",  
    exam = "Item Bank",  
    n_clones = 1,  
    max_items = NULL,  
    output_dir = NULL,  
    read_csv = TRUE  
)
```

## Arguments

<code>file_path</code>	Path to CSV or Excel item bank.
<code>provider</code>	LLM provider, such as "gemini", "openai", "groq", or "anthropic".
<code>model</code>	Provider model name.
<code>source_language</code>	Language of the source item bank.
<code>target_language</code>	Language for generated items.
<code>review_language</code>	Language for review comments.
<code>subject</code>	Subject name.
<code>exam</code>	Examination or item-bank name.
<code>n_clones</code>	Number of clones per source item.
<code>max_items</code>	Maximum number of valid source items to process. Use NULL to process all valid items.
<code>output_dir</code>	Optional output directory.
<code>read_csv</code>	If TRUE, returns the CSV output as a data frame.

**Value**

A data frame if read\_csv is TRUE; otherwise invisibly returns the CSV path.

---

aigra_image_models	<i>List supported AIGRA image models</i>
--------------------	--

---

**Description**

Returns the image model registry known to AIGRA. This function is CRAN-safe and does not call external APIs or require the Python backend.

**Usage**

```
aigra_image_models(provider = NULL)
```

**Arguments**

provider            Optional image provider, such as "gemini" or "openai".

**Value**

Supported image models.

---

aigra_latest_csv	<i>Get latest AIGRA CSV output path</i>
------------------	---

---

**Description**

Get latest AIGRA CSV output path

**Usage**

```
aigra_latest_csv()
```

**Value**

Path to latest CSV output.

---

aigra\_latest\_output     *Get latest AIGRA CSV output path*

---

**Description**

Get latest AIGRA CSV output path

**Usage**

```
aigra_latest_output()
```

**Value**

Path to the latest AIGRA CSV output.

---

aigra\_list\_outputs     *List AIGRA output files*

---

**Description**

Lists generated CSV and JSONL files from the backend output directory.

**Usage**

```
aigra_list_outputs()
```

**Value**

A data frame of output files.

---

aigra\_localize\_diagram\_prompts  
*Localize AIGRA diagram prompts*

---

**Description**

Adds a language instruction to diagram prompts so generated diagrams use the target language for visible labels.

**Usage**

```
aigra_localize_diagram_prompts(result, target_language = "English")
```

**Arguments**

result            A data frame returned by AIGRA.  
 target\_language    Target language for visible diagram text labels.

**Value**

Updated result data frame.

---

aigra\_multimodal\_template\_items    *Create AIGRA multimodal template items*

---

**Description**

Create AIGRA multimodal template items

**Usage**

```
aigra_multimodal_template_items(diagram_dir = NULL, create_diagrams = TRUE)
```

**Arguments**

diagram\_dir    Directory where source diagrams are stored or should be created.  
 create\_diagrams    If TRUE, creates sample source diagram PNG files.

**Value**

A data frame containing multimodal sample items.

---

aigra\_output\_dir    *AIGRA output directory*

---

**Description**

AIGRA output directory

**Usage**

```
aigra_output_dir()
```

**Value**

Path to the backend output directory.

---

aigra_outputs	<i>List AIGRA output files</i>
---------------	--------------------------------

---

**Description**

List AIGRA output files

**Usage**

```
aigra_outputs(pattern = "aigra(_tabular)?_results_.*\\.\\.(csv|jsonl)$")
```

**Arguments**

pattern            File-name pattern.

**Value**

A data frame of output files.

---

aigra_parse_items	<i>Parse assessment items from a PDF</i>
-------------------	--

---

**Description**

Parses a supported assessment-item PDF through the AIGRA Python backend and returns a data frame of source items.

**Usage**

```
aigra_parse_items(
  pdf_path = NULL,
  source_language = "Russian",
  subject = "Physics",
  exam = "Kazakhstan UNT"
)
```

**Arguments**

pdf\_path            Path to the source item-bank PDF. Defaults to backend data/Kz.pdf.  
source\_language     Language of the source item bank.  
subject             Subject name.  
exam                Examination name.

**Value**

A data frame of parsed assessment items.

---

aigra\_parse\_tabular\_items  
*Parse tabular item bank*

---

**Description**

Parses a CSV or Excel item bank directly from the supplied file path.

**Usage**

```
aigra_parse_tabular_items(  
  file_path,  
  source_language = "English",  
  subject = "General",  
  exam = "Item Bank"  
)
```

**Arguments**

file_path	Path to CSV or Excel item bank.
source_language	Default source language if the file does not contain source_language.
subject	Default subject if the file does not contain subject.
exam	Default exam name if the file does not contain exam.

**Value**

A data frame of parsed assessment items.

---

aigra\_plot\_diagram *Plot an AIGRA diagram image*

---

**Description**

Plot an AIGRA diagram image

**Usage**

```
aigra_plot_diagram(image_path)
```

**Arguments**

image_path	Path to a PNG image.
------------	----------------------

**Value**

Invisibly returns the image path.

---

aigra\_plot\_summary      *Plot AIGRA output quality summary*

---

**Description**

Creates a simple bar chart of item review statuses in an AIGRA output.

**Usage**

```
aigra_plot_summary(data = NULL)
```

**Arguments**

data                      Optional AIGRA output data frame. If NULL, reads the latest CSV output.

**Value**

Invisibly returns the status table used for plotting.

---

aigra\_print\_summary      *Print an AIGRA quality summary*

---

**Description**

Prints a compact quality summary for an AIGRA output.

**Usage**

```
aigra_print_summary(data = NULL)
```

**Arguments**

data                      Optional AIGRA output data frame. If NULL, reads the latest CSV output.

**Value**

Invisibly returns the summary list.

---

`aigra_print_validation`*Print tabular item-bank validation results*

---

**Description**

Print tabular item-bank validation results

**Usage**

```
aigra_print_validation(validation)
```

**Arguments**

`validation` Validation result returned by [aigra\\_validate\\_tabular\\_items\(\)](#).

**Value**

Invisibly returns the validation object.

---

`aigra_python_info`*Show AIGRA Python environment information*

---

**Description**

Show AIGRA Python environment information

**Usage**

```
aigra_python_info(backend_path = NULL)
```

**Arguments**

`backend_path` Path to the AIGRA\_BACKEND folder.

**Value**

A list with Python environment information.

---

aigra\_read\_latest\_csv *Read latest AIGRA CSV output*

---

**Description**

Read latest AIGRA CSV output

**Usage**

aigra\_read\_latest\_csv()

**Value**

A data frame containing the latest AIGRA CSV output.

---

aigra\_read\_latest\_output  
*Read latest AIGRA CSV output*

---

**Description**

Read latest AIGRA CSV output

**Usage**

aigra\_read\_latest\_output()

**Value**

A data frame containing the latest AIGRA CSV output.

---

aigra\_repair\_diagram\_prompts  
*Repair diagram prompts for diagram-dependent AIGRA results*

---

**Description**

Strengthens diagram prompts when a generated item requires a figure but the prompt is too vague for solving/review.

**Usage**

aigra\_repair\_diagram\_prompts(result)

**Arguments**

result            A data frame returned by AIGRA.

**Value**

Updated result data frame.

---

aigra\_set\_api\_keys    *Set 'AIGRA' API Keys*

---

**Description**

Sets API keys for supported providers from within R. These environment variables are inherited by the external 'Python' backend when generation functions are called.

**Usage**

```
aigra_set_api_keys(  
  gemini.API = NULL,  
  openai.API = NULL,  
  groq.API = NULL,  
  anthropic.API = NULL  
)
```

**Arguments**

gemini.API        Optional 'Gemini' API key.  
openai.API        Optional 'OpenAI' API key.  
groq.API          Optional 'Groq' API key.  
anthropic.API    Optional 'Anthropic' API key.

**Value**

Invisibly returns a named logical vector showing which keys are set.

**Examples**

```
keys <- aigra_set_api_keys()  
is.logical(keys)
```

---

aigra_set_backend	<i>Set the AIGRA Python backend</i>
-------------------	-------------------------------------

---

**Description**

Connects the R package to the AIGRA Python backend.

Sets the path to the AIGRA Python backend.

**Usage**

```
aigra_set_backend(backend_path = NULL)
```

```
aigra_set_backend(backend_path = NULL)
```

**Arguments**

backend_path	Path to the AIGRA_BACKEND folder. If NULL, uses the AIGRA_BACKEND_PATH environment variable.
--------------	--

**Value**

Invisibly returns backend configuration.

Invisibly returns the normalized backend path.

---

aigra_show_result_diagram
---------------------------

*Show a diagram from an AIGRA result row*

---

**Description**

Show a diagram from an AIGRA result row

**Usage**

```
aigra_show_result_diagram(result, row = 1)
```

**Arguments**

result	A data frame returned by AIGRA.
row	Row number to display.

**Value**

Invisibly returns the image path.

---

aigra_status	<i>Check AIGRA backend status</i>
--------------	-----------------------------------

---

**Description**

Checks whether the R package can access the Python backend.

**Usage**

```
aigra_status(backend_path = NULL)
```

**Arguments**

backend\_path    Optional path to the AIGRA\_BACKEND folder.

**Value**

A list containing backend path, Python path, Python version, and import status.

---

aigra_summarise_output	<i>Summarise an AIGRA output</i>
------------------------	----------------------------------

---

**Description**

Summarises the latest or supplied AIGRA CSV output.

**Usage**

```
aigra_summarise_output(data = NULL)
```

**Arguments**

data            Optional AIGRA output data frame. If NULL, reads the latest CSV output.

**Value**

A list with quality counts and simple rates.

---

aigra\_template\_items *Create an AIGRA tabular item-bank template*

---

**Description**

Creates a sample item-bank data frame with the required columns for CSV/Excel-based AIGRA generation.

**Usage**

```
aigra_template_items()
```

**Value**

A data frame containing sample item-bank rows.

---

aigra\_translate\_diagram\_prompts

*Translate/localize AIGRA diagram prompts using an LLM*

---

**Description**

This rewrites diagram prompts into the target language before image generation. Scientific symbols and units are preserved.

**Usage**

```
aigra_translate_diagram_prompts(  
  result,  
  target_language = "English",  
  provider = "gemini",  
  model = "gemini-3.1-pro-preview"  
)
```

**Arguments**

result	A data frame returned by AIGRA.
target_language	Target language for visible diagram labels.
provider	Text LLM provider.
model	Text LLM model.

**Value**

Updated result data frame.

---

`aigra_use_backend`      *Use AIGRA backend*

---

**Description**

Backward-compatible alias for `aigra_set_backend()`.

**Usage**

```
aigra_use_backend(backend_path = NULL)
```

**Arguments**

`backend_path`      Optional backend path. If NULL, uses `AIGRA_BACKEND_PATH`.

**Value**

Invisibly returns backend path.

---

`aigra_validate_tabular_items`  
*Validate a tabular AIGRA item bank*

---

**Description**

Checks a CSV or Excel item bank before parsing or generation.

**Usage**

```
aigra_validate_tabular_items(file_path)
```

**Arguments**

`file_path`      Path to CSV or Excel item bank.

**Value**

A list containing validation status, issues, and a summary.

---

`aigra_write_admin_html`*Write an AIGRA administration HTML file*

---

**Description**

Creates a print-ready HTML file containing generated item stems, diagrams, response options, and an answer key.

**Usage**

```
aigra_write_admin_html(  
    result,  
    file = NULL,  
    title = "AIGRA Generated Assessment Items",  
    include_key = TRUE,  
    include_metadata = FALSE,  
    only_accepted = TRUE  
)
```

**Arguments**

<code>result</code>	A data frame returned by AIGRA, preferably after diagram generation.
<code>file</code>	Output HTML file. If NULL, writes to backend outputs folder.
<code>title</code>	Title shown at the top of the paper.
<code>include_key</code>	If TRUE, include answer key at the end.
<code>include_metadata</code>	If TRUE, show topic, section, and difficulty.
<code>only_accepted</code>	If TRUE, include only rows with status ok or edited.

**Value**

Path to the HTML file.

---

`aigra_write_multimodal_template_excel`*Write an AIGRA multimodal Excel template*

---

**Description**

Creates source diagram PNGs and writes a ready-to-run Excel item bank.

**Usage**

```
aigra_write_multimodal_template_excel(
  file = NULL,
  diagram_dir = NULL,
  overwrite = FALSE
)
```

**Arguments**

file	Output Excel file path.
diagram_dir	Directory for source diagram PNGs.
overwrite	If TRUE, overwrite the Excel file.

**Value**

Path to the Excel file.

---

aigra_write_report	<i>Write an AIGRA HTML report</i>
--------------------	-----------------------------------

---

**Description**

Writes a CRAN-safe local HTML report from an AIGRA result data frame. If `diagram_path` is present, generated diagrams are shown inline.

**Usage**

```
aigra_write_report(result = NULL, file = NULL, title = "AIGRA Quality Report")
```

**Arguments**

result	A data frame returned by AIGRA. If NULL, reads latest output.
file	Output HTML path. If NULL, writes to backend outputs folder.
title	Report title.

**Value**

Path to the report.

---

`aigra_write_template_csv`*Write an AIGRA CSV item-bank template*

---

**Description**

Write an AIGRA CSV item-bank template

**Usage**

```
aigra_write_template_csv(file = "aigra_item_template.csv", overwrite = FALSE)
```

**Arguments**

<code>file</code>	Output CSV file path.
<code>overwrite</code>	If TRUE, overwrite an existing file.

**Value**

The normalized output file path.

---

`aigra_write_template_excel`*Write an AIGRA Excel item-bank template*

---

**Description**

Write an AIGRA Excel item-bank template

**Usage**

```
aigra_write_template_excel(  
  file = "aigra_item_template.xlsx",  
  overwrite = FALSE  
)
```

**Arguments**

<code>file</code>	Output Excel file path.
<code>overwrite</code>	If TRUE, overwrite an existing file.

**Value**

The normalized output file path.

---

ensure\_aigra\_python    *Ensure 'AIGRA' 'Python' Environment*

---

### Description

Creates or repairs the 'Python' virtual environment used by the 'AIGRA' backend.

### Usage

```
ensure_aigra_python(  
    backend_path = NULL,  
    python_version = "3.11",  
    force = FALSE,  
    install_providers = TRUE,  
    verbose = FALSE  
)
```

### Arguments

backend_path	Path to the 'AIGRA' backend folder. If NULL, the AIGRA_BACKEND_PATH environment variable is used.
python_version	'Python' version to use with 'uv'.
force	If TRUE, recreate the virtual environment.
install_providers	If TRUE, install provider packages.
verbose	If TRUE, show progress messages.

### Value

A list with 'Python' environment information.

### Examples

```
# This function creates or repairs a backend environment,  
# so the example only inspects the function interface.  
names(formals(ensure_aigra_python))
```

# Index

## \* utilities

- [aigra\\_generate\\_items, 8](#)
- [aigra\\_apply\\_diagram\\_agent, 3](#)
- [aigra\\_backend\\_help, 3](#)
- [aigra\\_backend\\_path, 4](#)
- [aigra\\_build\\_diagram\\_prompt, 4](#)
- [aigra\\_build\\_diagram\\_prompt\\_from\\_row, 5](#)
- [aigra\\_create\\_sample\\_source\\_diagrams, 5](#)
- [aigra\\_detect\\_diagram\\_required, 6](#)
- [aigra\\_example\\_item\\_bank, 6](#)
- [aigra\\_find\\_uv, 7](#)
- [aigra\\_generate\\_diagram, 7](#)
- [aigra\\_generate\\_diagram\\_fallback, 8](#)
- [aigra\\_generate\\_items, 8](#)
- [aigra\\_generate\\_multimodal\\_tabular\\_items, 10](#)
- [aigra\\_generate\\_result\\_diagrams, 11](#)
- [aigra\\_generate\\_result\\_diagrams\\_auto, 12](#)
- [aigra\\_generate\\_tabular\\_items, 13](#)
- [aigra\\_image\\_models, 14](#)
- [aigra\\_latest\\_csv, 14](#)
- [aigra\\_latest\\_output, 15](#)
- [aigra\\_list\\_outputs, 15](#)
- [aigra\\_localize\\_diagram\\_prompts, 15](#)
- [aigra\\_multimodal\\_template\\_items, 16](#)
- [aigra\\_output\\_dir, 16](#)
- [aigra\\_outputs, 17](#)
- [aigra\\_parse\\_items, 17](#)
- [aigra\\_parse\\_tabular\\_items, 18](#)
- [aigra\\_plot\\_diagram, 18](#)
- [aigra\\_plot\\_summary, 19](#)
- [aigra\\_print\\_summary, 19](#)
- [aigra\\_print\\_validation, 20](#)
- [aigra\\_python\\_info, 20](#)
- [aigra\\_read\\_latest\\_csv, 21](#)
- [aigra\\_read\\_latest\\_output, 21](#)
- [aigra\\_repair\\_diagram\\_prompts, 21](#)
- [aigra\\_set\\_api\\_keys, 22](#)
- [aigra\\_set\\_backend, 23](#)
- [aigra\\_set\\_backend\(\), 26](#)
- [aigra\\_show\\_result\\_diagram, 23](#)
- [aigra\\_status, 24](#)
- [aigra\\_summarise\\_output, 24](#)
- [aigra\\_template\\_items, 25](#)
- [aigra\\_translate\\_diagram\\_prompts, 25](#)
- [aigra\\_use\\_backend, 26](#)
- [aigra\\_validate\\_tabular\\_items, 26](#)
- [aigra\\_validate\\_tabular\\_items\(\), 20](#)
- [aigra\\_write\\_admin\\_html, 27](#)
- [aigra\\_write\\_multimodal\\_template\\_excel, 27](#)
- [aigra\\_write\\_report, 28](#)
- [aigra\\_write\\_template\\_csv, 29](#)
- [aigra\\_write\\_template\\_excel, 29](#)
- [ensure\\_aigra\\_python, 30](#)