Package: inferencer (via r-universe)

July 11, 2026

Type Package

Title Simple Unified Wrappers for Hosted Foundation Model Inference APIs

Version 0.1.4.5

Author Oliver Zhou [aut, cre]

Maintainer Oliver Zhou <oliver.yxzhou@gmail.com>

Description Provides lightweight R wrappers for querying and listing models from several hosted foundation model inference platforms, currently including Google Gemini, Groq, OpenRouter, Cerebras, and Ollama Cloud. The package is designed for simple inference workflows and quick experimentation, with minimal abstraction and a consistent interface across providers. It includes helper functions for model discovery, text generation, embeddings, image generation, and multimodal inputs, while leaving room for future support of provider-specific parameters and advanced options.

License MIT + file LICENSE

URL https://github.com/OliverLDS/inferencer

BugReports https://github.com/OliverLDS/inferencer/issues

Depends R (>= 4.1.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports data.table, httr2, jsonlite

Suggests knitr, rmarkdown, testthat

Config/testthat/edition 3

Config/pak/sysreqs libssl-dev

Repository https://oliverlds.r-universe.dev

Date/Publication 2026-07-03 06:39:23 UTC

RemoteUrl https://github.com/oliverlds/inferencer

RemoteRef HEAD

RemoteSha cf69bf0bb1de2e1e9f54c929017ba1558fdb7ed7

Contents

`embed_gemini` *Request Gemini Embeddings*

Description

Calls Gemini embedding endpoints for one or more text inputs.

Usage

```
embed_gemini(
  input,
  api_key = Sys.getenv("GEMINI_API_KEY"),
  model = "gemini-embedding-001",
  url0 = Sys.getenv("GEMINI_API_URL", unset =
```

```
    "https://generativelanguage.googleapis.com/v1beta/models"),
  task_type = NULL,
  title = NULL,
  output_dimensionality = NULL,
  json_list = FALSE
)
```

Arguments

input	A non-empty character vector.
api_key	Gemini API key. Defaults to `Sys.getenv("GEMINI_API_KEY")`.
model	Embedding model identifier.
url0	Base Gemini models URL.
task_type	Optional Gemini embedding `taskType`.
title	Optional embedding title, used with document-style inputs.
output_dimensionality	Optional embedding dimensionality override.
json_list	If `TRUE`, return the parsed JSON response as a list.

Value

A numeric matrix by default, or the parsed JSON response when `json_list = TRUE`.

---

embed_openrouter *Request OpenRouter Embeddings*

Description

Calls the OpenRouter embeddings endpoint for one or more text inputs.

Usage

```
embed_openrouter(
  input,
  model = "openai/text-embedding-3-small",
  dimensions = NULL,
  encoding_format = c("float", "base64"),
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = "https://openrouter.ai/api/v1/embeddings",
  json_list = FALSE
)
```

Arguments

input	A non-empty character vector.
model	Model identifier.
dimensions	Optional embedding dimensionality override.
encoding_format	Embedding encoding. Defaults to "float".
api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter embeddings endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A numeric matrix by default when encoding_format = "float", or the parsed JSON response when json_list = TRUE.

---

extract_openrouter_benchmarks

Extract Benchmark Scores from OpenRouter Model Metadata

Description

Scans OpenRouter model metadata for benchmark payloads, including possible Artificial Analysis benchmark fields when they are present in the models response.

Usage

```
extract_openrouter_benchmarks(
  models,
  benchmark_fields = c("benchmarks", "benchmark_scores", "artificial_analysis",
    "artificial_analysis_benchmarks", "benchmark_data")
)
```

Arguments

models	Parsed JSON from list_openrouter_models(json_list = TRUE), a list of OpenRouter model objects, or a data.frame returned by list_openrouter_models().
benchmark_fields	Candidate field names to inspect on each model.

Value

A data.table with one row per extracted benchmark metric. Returns an empty data.table when no benchmark fields are found.

`generate_image_gemini` *Generate Images with Gemini*

Description

Calls Gemini image-generation capable models and returns the generated image payloads as base64 strings.

Usage

```
generate_image_gemini(
  prompt,
  api_key = Sys.getenv("GEMINI_API_KEY"),
  model = "gemini-2.5-flash-image",
  url0 = Sys.getenv("GEMINI_API_URL", unset =
    "https://generativelanguage.googleapis.com/v1beta/models"),
  temperature = 0.7,
  top_p = 1,
  top_k = 40,
  max_tokens = NULL,
  response_modalities = c("TEXT", "IMAGE"),
  json_list = FALSE
)
```

Arguments

`prompt`	A non-empty character string.
`api_key`	Gemini API key. Defaults to `Sys.getenv("GEMINI_API_KEY")`.
`model`	Image-capable Gemini model identifier.
`url0`	Base Gemini models URL.
`temperature`	Sampling temperature.
`top_p`	Nucleus sampling parameter.
`top_k`	Top-k sampling parameter.
`max_tokens`	Optional maximum number of output tokens.
`response_modalities`	Modalities to request. Defaults to `c("TEXT", "IMAGE")`.
`json_list`	If `TRUE`, return the parsed JSON response as a list.

Value

A character vector of base64-encoded image payloads by default, or the parsed JSON response when `json_list = TRUE`.

`generate_image_openrouter` *Generate Images with OpenRouter*

Description

Calls the OpenRouter chat completions API for image-capable models and returns the first available image payload or URL.

Usage

```
generate_image_openrouter(
  prompt,
  model = "google/gemini-2.5-flash-image-preview",
  temperature = 0,
  top_p = 1,
  max_tokens = 2048L,
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = Sys.getenv("OPENROUTER_API_URL", unset =
    "https://openrouter.ai/api/v1/chat/completions"),
  json_list = FALSE
)
```

Arguments

`prompt`	A non-empty character string.
`model`	Model identifier.
`temperature`	Sampling temperature.
`top_p`	Nucleus sampling parameter.
`max_tokens`	Maximum number of output tokens.
`api_key`	OpenRouter API key. Defaults to `Sys.getenv("OPENROUTER_API_KEY")`.
`url`	OpenRouter chat completions endpoint.
`json_list`	If `TRUE`, return the parsed JSON response as a list.

Value

A character string containing an image URL or base64 payload by default, or the parsed JSON response when `json_list = TRUE`.

list_cerebras_models *List Cerebras Models*

Description

Retrieves available models from the public Cerebras models endpoint.

Usage

```
list_cerebras_models(
  url = "https://api.cerebras.ai/public/v1/models",
  json_list = FALSE
)
```

Arguments

url	Cerebras public models endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A data.table by default, or a parsed JSON list when json_list = TRUE.

list_gemini_models *List Gemini Models*

Description

Retrieves available models from the Gemini models endpoint.

Usage

```
list_gemini_models(
  api_key = Sys.getenv("GEMINI_API_KEY"),
  url = "https://generativelanguage.googleapis.com/v1beta/models",
  json_list = FALSE
)
```

Arguments

api_key	Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
url	Gemini models endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A data.table by default, or a parsed JSON list when json_list = TRUE.

list_groq_models *List Groq Models*

Description

Retrieves available models from the Groq models endpoint.

Usage

```
list_groq_models(
  api_key = Sys.getenv("GROQ_API_KEY"),
  url = "https://api.groq.com/openai/v1/models",
  json_list = FALSE
)
```

Arguments

api_key	Groq API key. Defaults to `Sys.getenv("GROQ_API_KEY")`.
url	Groq models endpoint.
json_list	If `TRUE`, return the parsed JSON response as a list.

Value

A `data.table` by default, or a parsed JSON list when `json_list = TRUE`.

list_ollama_models *List Ollama Cloud Models*

Description

Retrieves available models from the Ollama Cloud tags endpoint.

Usage

```
list_ollama_models(
  api_key = Sys.getenv("OLLAMA_API_KEY"),
  url = "https://ollama.com/api/tags",
  json_list = FALSE
)
```

Arguments

api_key	Ollama API key. Defaults to `Sys.getenv("OLLAMA_API_KEY")`.
url	Ollama Cloud tags endpoint.
json_list	If `TRUE`, return the parsed JSON response as a list.

Value

A `data.table` by default, or a parsed JSON list when `json_list = TRUE`.

`list_openrouter_audio_models`

List OpenRouter Audio Models

Description

Filters the general OpenRouter models catalog to models with audio input or output modalities.

Usage

```
list_openrouter_audio_models(
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = "https://openrouter.ai/api/v1/models",
  json_list = FALSE
)
```

Arguments

`api_key`	OpenRouter API key. Defaults to `Sys.getenv("OPENROUTER_API_KEY")`.
`url`	OpenRouter models endpoint.
`json_list`	If `TRUE`, return the filtered parsed JSON response as a list.

Value

A filtered `data.table` by default, or a filtered parsed JSON list when `json_list = TRUE`.

`list_openrouter_embedding_models`

List OpenRouter Embedding Models

Description

Filters the general OpenRouter models catalog to models with embedding output modalities.

Usage

```
list_openrouter_embedding_models(
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = "https://openrouter.ai/api/v1/models",
  json_list = FALSE
)
```

Arguments

api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter models endpoint.
json_list	If TRUE, return the filtered parsed JSON response as a list.

Value

A filtered data.table by default, or a filtered parsed JSON list when json_list = TRUE.

---

list_openrouter_image_models
List OpenRouter Image Models

Description

Filters the general OpenRouter models catalog to models with image output modalities.

Usage

```
list_openrouter_image_models(
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = "https://openrouter.ai/api/v1/models",
  json_list = FALSE
)
```

Arguments

api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter models endpoint.
json_list	If TRUE, return the filtered parsed JSON response as a list.

Value

A filtered data.table by default, or a filtered parsed JSON list when json_list = TRUE.

`list_openrouter_models` *List OpenRouter Models*

Description

Retrieves available models from the OpenRouter models endpoint.

Usage

```
list_openrouter_models(
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = "https://openrouter.ai/api/v1/models",
  json_list = FALSE
)
```

Arguments

`api_key`	OpenRouter API key. Defaults to `Sys.getenv("OPENROUTER_API_KEY")`.
`url`	OpenRouter models endpoint.
`json_list`	If `TRUE`, return the parsed JSON response as a list.

Value

A `data.table` by default, or a parsed JSON list when `json_list = TRUE`.

`list_openrouter_multimodal_models` *List OpenRouter Multimodal Models*

Description

Filters the general OpenRouter models catalog to models that support multiple input modalities.

Usage

```
list_openrouter_multimodal_models(
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = "https://openrouter.ai/api/v1/models",
  json_list = FALSE
)
```

Arguments

api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter models endpoint.
json_list	If TRUE, return the filtered parsed JSON response as a list.

Value

A filtered data.table by default, or a filtered parsed JSON list when json_list = TRUE.

---

list_openrouter_video_models

List OpenRouter Video Generation Models

Description

Retrieves available video generation models and their normalized feature metadata from the OpenRouter video models endpoint.

Usage

```
list_openrouter_video_models(
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = "https://openrouter.ai/api/v1/videos/models",
  json_list = FALSE
)
```

Arguments

api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter video models endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A data.table by default, or a parsed JSON list when json_list = TRUE.

query_cerebras *Query a Cerebras Chat Model*

Description

Sends a single user prompt to the Cerebras chat completions API.

Usage

```
query_cerebras(
  prompt,
  model = c("gpt-oss-120b", "zai-glm-4.7", "llama3.1-8b",
    "qwen-3-235b-a22b-instruct-2507"),
  api_key = Sys.getenv("CEREBRAS_API_KEY"),
  url = "https://api.cerebras.ai/v1/chat/completions",
  json_list = FALSE
)
```

Arguments

prompt	A non-empty character string.
model	Model identifier.
api_key	Cerebras API key. Defaults to `Sys.getenv("CEREBRAS_API_KEY")`.
url	Cerebras chat completions endpoint.
json_list	If `TRUE`, return the parsed JSON response as a list.

Value

A character string by default, or a parsed JSON list when `json_list = TRUE`.

See Also

`list_cerebras_models()`

query_fallback *Query Providers with Ordered Fallback*

Description

Tries the package's text-query wrappers in order until one succeeds: `query_gemini()`, then `query_openrouter()`, then `query_groq()`.

Usage

```
query_fallback(
  prompt,
  json_list = FALSE,
  api_key_gemini = Sys.getenv("GEMINI_API_KEY"),
  api_key_openrouter = Sys.getenv("OPENROUTER_API_KEY"),
  api_key_groq = Sys.getenv("GROQ_API_KEY")
)
```

Arguments

prompt	A non-empty character string.
json_list	If TRUE, return a list containing the successful provider name and parsed response object.
api_key_gemini	Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
api_key_openrouter	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
api_key_groq	Groq API key. Defaults to Sys.getenv("GROQ_API_KEY").

Value

A character string by default. When json_list = TRUE, returns a list with elements provider and response.

---

query_gemini *Query a Gemini Model*

Description

Sends a single user prompt to the Gemini generateContent API.

Usage

```
query_gemini(
  prompt,
  api_key = Sys.getenv("GEMINI_API_KEY"),
  model = "gemini-2.5-flash",
  url0 = Sys.getenv("GEMINI_API_URL", unset =
    "https://generativelanguage.googleapis.com/v1beta/models"),
  temperature = 0.7,
  top_p = 1,
  top_k = 40,
  max_tokens = NULL,
  response_modalities = NULL,
  speech_config = NULL,
  json_list = FALSE
)
```

Arguments

prompt	A non-empty character string.
api_key	Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
model	Model identifier.
url0	Base Gemini models URL.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
top_k	Top-k sampling parameter.
max_tokens	Optional maximum number of output tokens.
response_modalities	Optional response modalities, for example c("TEXT") or c("AUDIO").
speech_config	Optional Gemini speechConfig object supplied as an R list.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default. For audio responses, returns the base64-encoded audio data from inlineData$data. Returns the parsed JSON list when json_list = TRUE.

---

query_gemini_content *Query a Gemini Multimodal Model*

Description

Sends Gemini generateContent requests using either a simple text prompt or an explicit list of content parts. This allows text, image, audio, PDF, and other supported non-text inputs to be passed through the same wrapper.

Usage

```
query_gemini_content(
  prompt = NULL,
  parts = NULL,
  api_key = Sys.getenv("GEMINI_API_KEY"),
  model = "gemini-2.5-flash",
  url0 = Sys.getenv("GEMINI_API_URL", unset =
    "https://generativelanguage.googleapis.com/v1beta/models"),
  temperature = 0.7,
  top_p = 1,
  top_k = 40,
  max_tokens = NULL,
  response_modalities = NULL,
  speech_config = NULL,
  json_list = FALSE
)
```

Arguments

prompt	A non-empty character string. Ignored when parts is supplied.
parts	Optional Gemini parts payload supplied as a list. Use this for multimodal inputs such as inlineData or fileData.
api_key	Gemini API key. Defaults to Sys.getenv("GEMINI_API_KEY").
model	Model identifier.
url0	Base Gemini models URL.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
top_k	Top-k sampling parameter.
max_tokens	Optional maximum number of output tokens.
response_modalities	Optional response modalities, for example c("TEXT"), c("AUDIO"), or c("TEXT", "IMAGE").
speech_config	Optional Gemini speechConfig object supplied as an R list.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string when Gemini returns text, a base64 string when the first returned part is inline binary data, or the parsed JSON response when json_list = TRUE.

---

query_groq *Query a Groq Chat Model*

Description

Sends a single user prompt to the Groq chat completions API.

Usage

```
query_groq(
  prompt,
  api_key = Sys.getenv("GROQ_API_KEY"),
  url = Sys.getenv("GROQ_API_URL", unset =
    "https://api.groq.com/openai/v1/chat/completions"),
 model = c("groq/compound", "allam-2-7b", "groq/compound-mini", "qwen/qwen3-32b",
  "openai/gpt-oss-20b", "canopylabs/orpheus-v1-english", "openai/gpt-oss-120b",
  "whisper-large-v3", "llama-3.3-70b-versatile", "moonshotai/kimi-k2-instruct-0905",
   "whisper-large-v3-turbo", "meta-llama/llama-prompt-guard-2-86m",
   "moonshotai/kimi-k2-instruct", "meta-llama/llama-prompt-guard-2-22m",
  "meta-llama/llama-4-scout-17b-16e-instruct", "openai/gpt-oss-safeguard-20b",
   "llama-3.1-8b-instant", "canopylabs/orpheus-arabic-saudi"),
```

```
  temperature = 1,
  top_p = 1,
  max_tokens = 1024,
  stream = FALSE,
  json_list = FALSE
)
```

Arguments

prompt	A non-empty character string.
api_key	Groq API key. Defaults to Sys.getenv("GROQ_API_KEY").
url	Groq chat completions endpoint.
model	Model identifier.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
max_tokens	Maximum number of output tokens.
stream	Whether to request streaming output.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default, or a parsed JSON list when json_list = TRUE.

---

query_ollama *Query an Ollama Cloud Chat Model*

Description

Sends a single user prompt to the Ollama Cloud chat API.

Usage

```
query_ollama(
  prompt,
  model = "gpt-oss:120b",
  api_key = Sys.getenv("OLLAMA_API_KEY"),
  url = "https://ollama.com/api/chat",
  json_list = FALSE
)
```

Arguments

prompt	A non-empty character string.
model	Model identifier.
api_key	Ollama API key. Defaults to Sys.getenv("OLLAMA_API_KEY").
url	Ollama Cloud chat endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default, or a parsed JSON list when `json_list = TRUE`.

`query_openrouter` *Query an OpenRouter Chat Model*

Description

Sends a single user prompt to the OpenRouter chat completions API.

Usage

```
query_openrouter(
  prompt,
  model = "openrouter/free",
  temperature = 0,
  top_p = 1,
  max_tokens = 2048L,
  reasoning = TRUE,
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = Sys.getenv("OPENROUTER_API_URL", unset =
    "https://openrouter.ai/api/v1/chat/completions"),
  json_list = FALSE
)
```

Arguments

`prompt`	A non-empty character string.
`model`	Model identifier.
`temperature`	Sampling temperature.
`top_p`	Nucleus sampling parameter.
`max_tokens`	Maximum number of output tokens.
`reasoning`	Whether to enable reasoning mode.
`api_key`	OpenRouter API key. Defaults to `Sys.getenv("OPENROUTER_API_KEY")`.
`url`	OpenRouter chat completions endpoint.
`json_list`	If `TRUE`, return the parsed JSON response as a list.

Value

A character string by default, or a parsed JSON list when `json_list = TRUE`.

query_openrouter_content *Query an OpenRouter Multimodal Model*

Description

Sends a single OpenRouter chat-completions request using either simple text content or an explicit multimodal content block list.

Usage

```
query_openrouter_content(
  content,
  model = "openrouter/free",
  temperature = 0,
  top_p = 1,
  max_tokens = 2048L,
  reasoning = TRUE,
  modalities = NULL,
  api_key = Sys.getenv("OPENROUTER_API_KEY"),
  url = Sys.getenv("OPENROUTER_API_URL", unset =
    "https://openrouter.ai/api/v1/chat/completions"),
  json_list = FALSE
)
```

Arguments

content	A non-empty character string or a non-empty list of OpenAI style content blocks.
model	Model identifier.
temperature	Sampling temperature.
top_p	Nucleus sampling parameter.
max_tokens	Maximum number of output tokens.
reasoning	Whether to enable reasoning mode.
modalities	Optional output modalities, for example "text" or c("text", "image").
api_key	OpenRouter API key. Defaults to Sys.getenv("OPENROUTER_API_KEY").
url	OpenRouter chat completions endpoint.
json_list	If TRUE, return the parsed JSON response as a list.

Value

A character string by default when the assistant returns text, the first image payload or URL when text is absent but images are returned, or the parsed JSON response when json_list = TRUE.

shell-scripts *Shell Script Helpers*

Description

inferencer also ships a small optional shell layer under `inst/shell`. These scripts are executable `zsh` command-line helpers that mirror common R wrapper workflows: `query_gemini`, `query_groq`, `query_openrouter`, `query_ollama`, `query_fallback`, provider-specific `list_*_models` scripts, and `render_markdown_terminal`.

Details

The shell scripts read API keys and defaults from shell environment variables, typically configured in `.zprofile`. The variable names are aligned with the R wrappers, such as `GEMINI_API_KEY`, `GROQ_API_KEY`, `OPENROUTER_API_KEY`, and `OLLAMA_API_KEY`.

They are intentionally minimal and depend on external `curl` and `jq` binaries. Query scripts print the model response text by default and return the full parsed JSON payload when called with `--json`. Pipe query output to `render_markdown_terminal` when terminal markdown rendering is desired. `query_fallback` tries `query_gemini`, then `query_openrouter`, then `query_groq` using each script's default model.

write_gemini_audio *Write Gemini Audio to a File*

Description

Decodes base64 audio returned by `query_gemini()` and writes it as raw PCM or a WAV file.

Usage

```
write_gemini_audio(
  x,
  path,
  format = c("pcm", "wav"),
  sample_rate = 24000L,
  channels = 1L,
  bits_per_sample = 16L
)
```

Arguments

`x`	A base64-encoded audio string, or a parsed JSON response returned by `query_gemini(..., json_list = TRUE)`.
`path`	Output file path.

format	Output format: "pcm" or "wav".
sample_rate	Sample rate used when writing WAV output.
channels	Number of audio channels used when writing WAV output.
bits_per_sample	Bit depth used when writing WAV output.

Value

Invisibly returns path.

Index