

# MGP API Usage Guide

## Introduction

The **Multivariate Genotype-Phenotype (MGP) Analysis** allows you to explicitly model the joint effects of biologically coherent collections of genes on a multivariate trait—craniofacial shape. This method leverages biological process Gene Ontology (GO) annotations to select skeletal and facial development gene sets and solves for the axis of shape variation that maximally covaries with gene set marker variation.

## Base URL

`http://10.44.121.196:6238/`

You can also explore and test the API documentation by visiting the following URL:

`http://10.44.121.196:6238/__docs__/`

## Endpoints

- **Standard MGP:** `/mgp`
- **Custom MGP:** `/custom_mgp` (Takes a gene list)

## Request Method

**GET** requests should be used for both the `/mgp` and `/custom_mgp` endpoints.

## Request Format

Send a **GET** request with a JSON body containing the following parameters:

```
{
  "GO.term": "chondrocyte differentiation",
  "lambda": 0,
  "pls_axis": 1,
  "pheno": "A_lm_gen_sex_size",
  "permutation": 0,
  "doPermutation": false,
```

```

    "rangenes": 3,
    "doRangenes": false,
    "pheno_index": "1:65",
    "remove_pc1": false,
    "use_standardized_PCA": false
}

```

For the **Custom MGP** endpoint (`/custom_mgp`), replace the `GO.term` with a **gene list**, which must be a comma-separated list of gene names. Example:

```

{
  "genelist": "Bmp7, Bmp2, Bmp4, Ankrd11",
  "lambda": 0,
  "pls_axis": 1,
  "pheno": "A_lm_gen_sex_size",
  "permutation": 0,
  "doPermutation": false,
  "rangenes": 3,
  "doRangenes": false,
  "pheno_index": "1:65",
  "remove_pc1": false,
  "use_standardized_PCA": false
}

```

## Parameter Options

- **GO.term** (for Standard MGP): Any valid Gene Ontology term.
- **genelist** (for Custom MGP): A comma-separated list of gene names, e.g., "Bmp7, Bmp2, Bmp4, Ankrd11".
- **lambda**: Numeric value (typically between 0 and 1).
- **pls\_axis**: Integer value (typically 1-3).
- **pheno**: Choose from the following options:
  - "Y" – Original data from the old paper.
  - "A\_lm\_raw" – All mice, landmarks data (not normalized with GPA).
  - "A\_lm\_gen" – All mice, generation regressed out.
  - "F\_lm\_gen" – Female mice, generation regressed out.
  - "M\_lm\_gen" – Male mice, generation regressed out.
  - "A\_lm\_gen\_sex" – All mice, generation and sex regressed out.
  - "A\_lm\_gen\_sex\_size" – All mice, generation, sex, and size regressed out.
  - "A\_lm\_gen\_size" – All mice, generation and size regressed out.
  - "F\_lm\_gen\_size" – Female mice, generation and size regressed out.

- "M\_lm\_gen\_size" – Male mice, generation and size regressed out.
  - "A\_lm" – All mice, normalized data.
  - "F\_lm" – Female mice, normalized data.
  - "M\_lm" – Male mice, normalized data.
  - "YHuman" – Original human data.
  - "YHumanTANZ" – Human data (TANZ dataset).
  - "YHumanDense" – Dense human landmarks data.
  - "YHumanTANZDense" – Dense human landmarks data (TANZ dataset).
- **permutation:** Integer value indicating the number of permutations.
  - **doPermutation:** Boolean indicating whether to perform permutations.
  - **rangenes:** Integer value for the range of genes to analyze.
  - **doRangenes:** Boolean indicating whether to analyze a range of genes.
  - **pheno\_index:** String specifying the phenotype index. The value depends on the **pheno** chosen:
    - If **pheno** is "Y":
      - \* **Phenotype Index (All Landmarks):** "1:54"
    - If **pheno** is one of the **lmsGenOptions** (e.g., "A\_lm\_gen\_sex\_size", "F\_lm\_gen\_size"):
      - \* **Phenotype Index (All Landmarks):** "1:748"
      - \* **Phenotype Index (Original Landmarks):** "1:93"
    - If **pheno** is one of the **lmsHumanOptions** (e.g., "YHuman", "YHumanDense"):
      - \* **Phenotype Index (Sparse Landmarks):** "1:65"
      - \* **Phenotype Index (Dense Landmarks):** "1:5629"
  - **remove\_pc1:** Boolean indicating whether to remove the first principal component.
  - **use\_standardized\_PCA:** Boolean indicating whether to use standardized PCA.

## Example Usage with R Code

Here is an example of how to make a request to the MGP API using R:

```
# Load necessary libraries
library(httr)
library(jsonlite)

# Define the base URL and endpoint
base_url <- "http://10.44.121.196:6238/"
endpoint <- "/mgp"

# Create the request body as a list
request_body <- list(
```

```

GO.term = "chondrocyte differentiation",
lambda = 0,
pls_axis = 1,
pheno = "A_lm_gen_sex_size",
permutation = 0,
doPermutation = FALSE,
rangenes = 3,
doRangenes = FALSE,
pheno_index = "1:65",
remove_pc1 = FALSE,
use_standardized_PCA = FALSE
)

# Convert the request body to JSON format
json_body <- toJSON(request_body, auto_unbox = TRUE)

# Make the GET request
response <- GET(
  url = paste0(base_url, endpoint),
  body = json_body,
  encode = "json",
  content_type_json()
)

# Parse the response
response_content <- content(response, "text", encoding = "UTF-8")
parsed_response <- fromJSON(response_content)

# Print the parsed response
print(parsed_response)

```

In this example, the R code uses the `httr` package to send a GET request to the MGP API. The request body is created in R as a list, converted to JSON, and sent to the API. The response is parsed and printed out in R.