

# Documentación Programa de Morfometría Matricial en R

---

## 1. Introducción

El programa en R constituye un entorno completo de análisis morfométrico. Su estructura está organizada en módulos que permiten importar, procesar, normalizar, y analizar datos provenientes de contornos o perfiles topológicos. Cada sección cumple una función específica dentro del flujo de análisis.

## 1. Descripción General

Este programa procesa, analiza y visualiza contornos morfométricos de objetos geométricos. Sus principales funcionalidades son:

1. Importación de contornos internos y externos desde archivos `.txt`.
2. Cálculo de métricas morfométricas: longitud de contorno, ángulo, volumen.
3. Normalización y ajuste de contornos al origen cero.
4. Generación de arrays y listas para análisis multivariante.
5. Análisis exploratorio: estadísticos básicos, correlación, covarianza, PCA.
6. Visualización 2D y 3D de contornos y comparaciones entre individuos.
7. Cálculo de distancias entre coordenadas y áreas bajo curvas.

## **2. Estructura General**

El código está dividido en secciones I a VI. A continuación, se describe el propósito de cada una:

### **I – Importación de datos**

(IMP): Lee archivos de contornos, calcula la longitud acumulada (LC), ángulos (A) y volúmenes (V).

### **II – Conversión a matriz**

(AR): Convierte los datos en un array tridimensional para análisis vectorizado.

### **III – Normalización y orientación**

(NF, OR\_ci, OR\_ce): Alinea los contornos para hacerlos comparables entre individuos.

### **IV – Funciones de comparación**

(IND, COMP, PER\_1/2/3): Permite visualizar y comparar curvas o individuos.

### **V – Análisis exploratorio**

(sumario): Calcula estadísticas descriptivas por variable e individuo.

### **VI – Análisis multivariante**

(COR, COV, PCA, COPO, AREA.FUN): Realiza análisis de correlaciones, covarianzas, PCA, distancias morfométricas y cálculo de áreas.

## 3. Estructura del Programa

### 3.1 librerias.R

**Propósito:** Gestión de dependencias y carga de librerías.

Funciones principales:

`install_if_missing(pkg)`: Instala una librería si no está presente.

Carga librerías esenciales: `bezier`, `plotrix`, `scatterplot3d`, `tcltk`, `stats`, `pracma`, `gWidgets2`, `gWidgets2tcltk`, `digest`.

### 3.2 morfometria.R

**Propósito:** Procesamiento de contornos y métricas morfométricas.

Funciones clave:

`IMP()`: Importa archivos de contornos y calcula longitud, ángulo y volumen.

`AR()`: Genera arrays interpolados con curvas Bezier.

`NF()`: Normaliza valores de X, Y, CL, Ángulo y Volumen.

`OR_ce()`, `OR_ci()`: Ajustan contornos al origen cero.

`IND()`, `GT()`: Identificación de individuos y tablas de datos.

Visualización: `COMP()`, `PER_1()`, `PER_2()`, `PER_3()`, `P3D()`, `PO()`.

### 3.3 analisis.R

**Propósito:** Análisis exploratorio y multivariante.

Funciones clave:

`sumario()`: Resumen estadístico (mínimo, cuartiles, mediana, media, máximo, sd).

`COR()`, `COV()`: Matrices de correlación y covarianza.

`PCA()`: Análisis de componentes principales (prcomp/princomp).

`COP0()`: Calcula distancias euclidianas entre coordenadas.

`AREA.FUN()`: Calcula el área bajo curvas con integración trapezoidal.

## 4. Detalle del procedimiento

### 4.1 Sección I – Importación de datos (IMP)

La función IMP(path) lee archivos de texto que contienen coordenadas X e Y de contornos. Para cada archivo, calcula:

- Longitud acumulada (LC):  $LC = \sum \sqrt{(\Delta X)^2 + (\Delta Y)^2}$
- Ángulo local (A):  $A = \arccos(\Delta X / \sqrt{(\Delta X)^2 + (\Delta Y)^2}) * 180/\pi$
- Volumen estimado (V): basado en rotación del perfil, con  $F = X^2\pi Y$

Devuelve una lista con las variables data, Y, CL, A y V.

### 4.2 Sección II – Conversión a matriz (AR)

Convierte las listas de contornos en una estructura matricial (array tridimensional) con dimensiones [n\_puntos, 5, n\_individuos], correspondiente a las variables data, Y, CL, A y V.

### 4.3 Sección III – Normalización y orientación (NF, OR\_ci, OR\_ce)

Estas funciones ajustan los contornos a un origen común, eliminan diferencias de escala y permiten comparar morfologías de forma estandarizada.

### 4.4 Sección IV – Funciones de comparación (IND, COMP, PER\_1/2/3)

Permiten identificar individuos, comparar contornos internos y externos, y graficar las distintas variables de forma conjunta o separada.

### 4.5 Sección V – Análisis exploratorio (sumario)

La función sumario(data, object) produce resúmenes estadísticos para cada variable e individuo. Calcula: Mínimo, Primer cuartil, Mediana, Media, Tercer cuartil, Máximo y Desviación estándar. Organiza los resultados en un array tridimensional: [estadístico, variable, individuo].

## 4.6 Sección VI – Análisis multivariante

Incluye varias funciones analíticas:

- **COR()**: calcula matrices de correlación entre variables.
- **COV()**: calcula matrices de covarianza.
- **PCA()**: realiza análisis de componentes principales (ACP) mediante prcomp o princomp.
- **COPO()**: calcula distancias morfométricas punto a punto entre individuos.
- **AREA.FUN()**: integra el área bajo la curva usando el método trapezoidal (trapz).

## 5. Flujo de Trabajo (Ejemplo Práctico)

### # 1. Importar librerías y dependencias

```
source("librerias.R")
```

### # 2. Importar contornos

```
L1 <- IMP(1, ruta) # Contornos internos
```

```
L2 <- IMP(2, ruta) # Contornos externos
```

### # 3. Normalizar datos

```
L3 <- NF(L1)
```

```
L4 <- NF(L2)
```

### # 4. Traslación de contornos al origen cero (opcional)

```
L5 <- OR_ci(L1, L2)
```

```
L6 <- OR_ce(L2)
```

## **# 5. Generación de arrays interpolados**

```
A1 <- AR(1, 20, ruta) # 20 puntos arbitrarios
```

```
A2 <- AR(2, 20, ruta)
```

```
A3 <- NF(A1)
```

```
A4 <- NF(A2)
```

## **# 6. Plantilla de individuos**

```
ind <- IND(A1, destination)
```

## **# 7. Análisis exploratorio**

```
S1 <- sumario(L1)
```

```
S2 <- sumario(A1)
```

```
CO1 <- COR(L1)
```

```
CO2 <- COR(A1)
```

```
CV1 <- COV(L1)
```

```
CV2 <- COV(A1)
```

## **# 8. Distancias entre coordenadas**

```
DC <- COPO(A3)
```

## **# 9. Análisis de componentes principales**

```
ACP1 <- PCA(L1, 1, TRUE)
```

```
ACP2 <- PCA(A1, 1, TRUE)
```

```
ACP3 <- PCA(CO1, 1, FALSE)
```

```
ACP4 <- PCA(CO2, 1, FALSE)
```

## # 10. Visualización (ejemplos)

AREA.FUN(A3, 1)

COMP(L4, L3, 1, 2)

PER\_1(L2, L1, 1, 2, 3, 4)

P3D(L1, 1)

PO(A3, 1, 1)

## 4. Notas y Recomendaciones

1. Todas las funciones manejan **listas** y **arrays 3D**, permitiendo múltiples individuos y variables.
2. Las rutas (**ruta** y **destination**) deben definirse correctamente antes de ejecutar el flujo de trabajo.
3. La interpolación con Bezier y normalización facilitan comparaciones entre individuos de distinto tamaño.
4. Se recomienda revisar las funciones de visualización antes de procesar grandes conjuntos de datos para optimizar rendimiento.