

# Imaging, shape, and volumetric workflows

## Contents

|     |                                                   |   |
|-----|---------------------------------------------------|---|
| 0.1 | Wide-image scalar regression screen . . . . .     | 1 |
| 0.2 | Latent image-to-image regression screen . . . . . | 2 |
| 0.3 | Ordered-contour shape PCA . . . . .               | 3 |
| 0.4 | Volumetric ISPAT-3D input . . . . .               | 4 |

This vignette uses three compact deterministic datasets: 48 image subjects, six contours, and 72 volumetric observations.

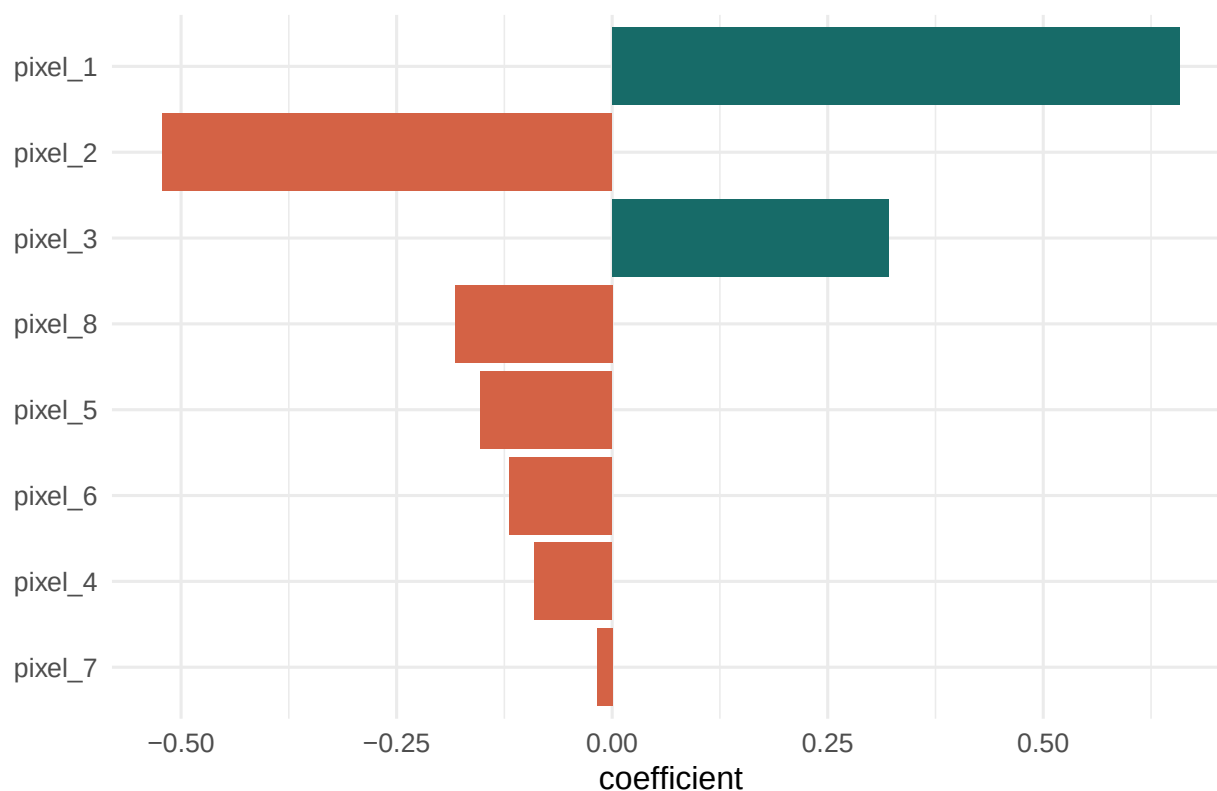
### 0.1 Wide-image scalar regression screen

```
image <- demo_image_data(n = 48, features = 8, seed = 2026)
pixel_features <- grep("^pixel_", names(image), value = TRUE)

scalar <- run_analysis(
  "scalar_image", image, list(outcome = "outcome"), pixel_features,
  list(image_ridge = 5, seed = 2026)
)
scalar$summary
#>      subjects image_features      ridge heldout_rmse heldout_mae
#>    48.0000000    8.0000000    5.0000000    0.6494585    0.5956614
head(scalar$table)
#>  feature coefficient
#> 1 pixel_1    0.6577132
#> 2 pixel_2   -0.5211847
#> 3 pixel_3    0.3212493
#> 8 pixel_8   -0.1822060
#> 5 pixel_5   -0.1525856
#> 6 pixel_6   -0.1187539

scalar$plot
```

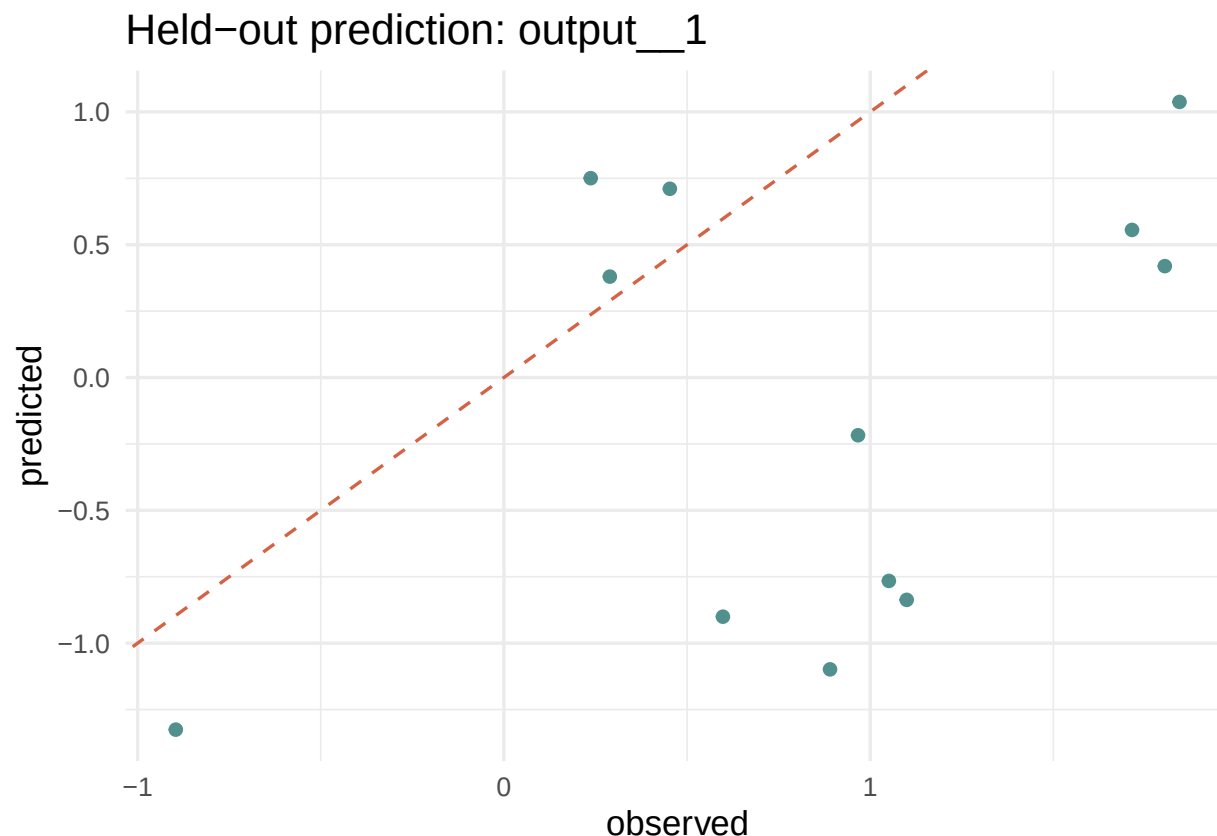
Largest standardized image–feature coefficients



## 0.2 Latent image-to-image regression screen

```
paired_features <- grep("^(input_|output_)", names(image), value = TRUE)
latent <- run_analysis(
  "image_to_image", image, list(), paired_features,
  list(latent_factors = 3L, seed = 2026)
)
latent$summary
#>      subjects      input_features      output_features
#>      48.000000      8.000000      8.000000
#>      latent_factors overall_heldout_rmse
#>      3.000000      0.9418131
head(latent$table)
#>      output_feature heldout_rmse
#> output__1      output__1      1.2612515
#> output__2      output__2      0.7186775
#> output__3      output__3      1.2653351
#> output__4      output__4      0.6910500
#> output__5      output__5      0.5696722
#> output__6      output__6      1.0008603
```

```
latent$plot
```



These fast screens expect already aligned and consistently ordered image features. They are not the full SV-NN, ST-CAR, or SBLF posterior engines.

```
method_citations("scalar_image")
#> [1] "Bhadury, S. (2026). SpatialMethodsWorkbench: Spatial Methods Workbench (Version 0.2.1) [Compute
#> [2] "Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal P
#> [3] "Wu, B., Wu, K., and Kang, J. (2025). Bayesian Scalar-on-Image Regression with a Spatially Varyi
#> [4] "Xu, Y. and Kang, J. (2025). Bayesian Image Regression with Soft-thresholded Conditional Autoreg
method_citations("image_to_image")
#> [1] "Bhadury, S. (2026). SpatialMethodsWorkbench: Spatial Methods Workbench (Version 0.2.1) [Compute
#> [2] "Guo, C., Kang, J., and Johnson, T. D. (2022). A Spatial Bayesian Latent Factor Model for Image-
```

### 0.3 Ordered-contour shape PCA

```
contours <- demo_contour_data(contours = 6, points = 16, seed = 2026)
head(contours)
#>   contour_id order      x      y
#> 1         C01     1 1.023845e+00 0.4031477
#> 2         C01     2 7.602709e-01 0.7227267
#> 3         C01     3 3.686349e-01 0.8460146
#> 4         C01     4 5.897198e-17 0.9155559
#> 5         C01     5 -3.841713e-01 0.8816706
#> 6         C01     6 -7.758343e-01 0.7375215

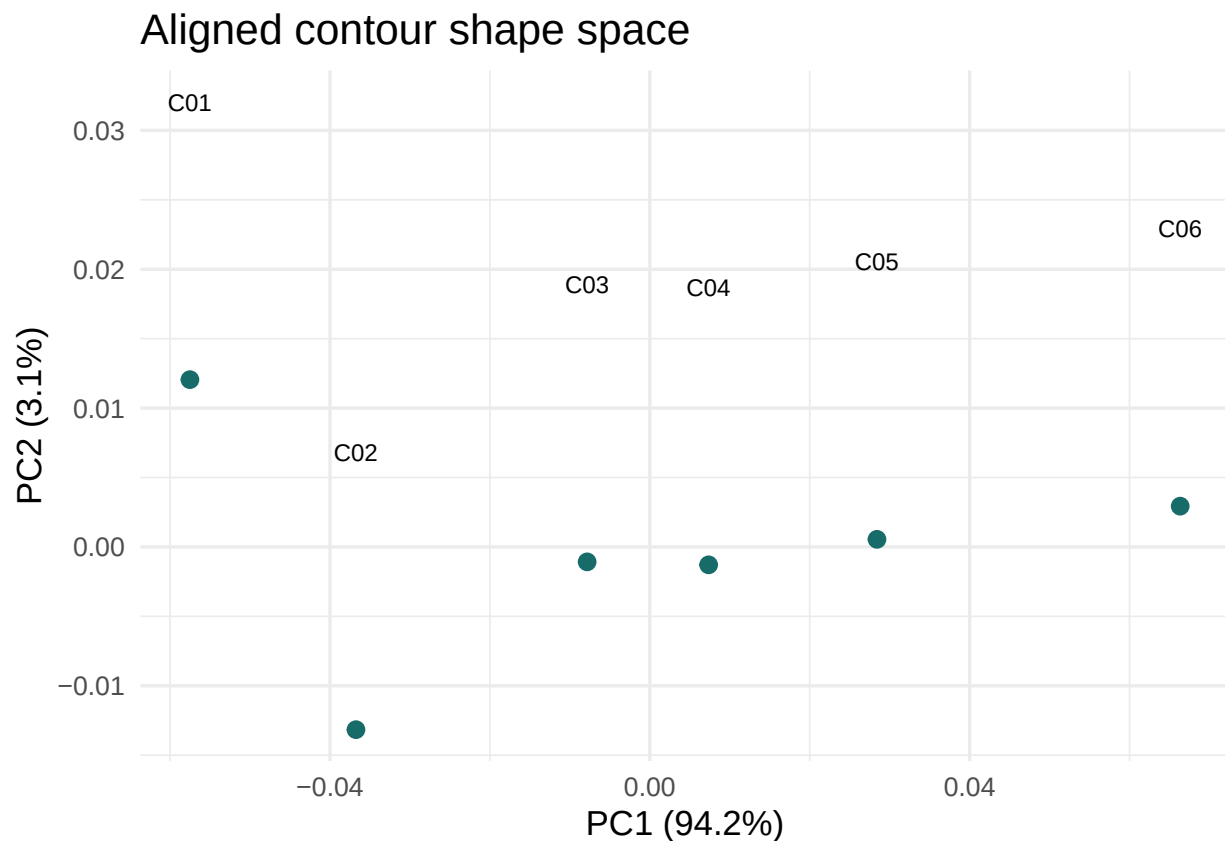
shape <- run_analysis(
  "shape_pca", contours,
```

```

list(id = "contour_id", x = "x", y = "y"), character(),
list(landmarks = 16L)
)
shape$summary
#>   contours landmarks pc1_variance pc2_variance
#>   6.000000   16.000000    94.150808    3.113624
shape$table
#>   id      PC1      PC2
#> C01 C01 -0.057519448  0.0120477771
#> C02 C02 -0.036752447 -0.0131625653
#> C03 C03 -0.007838020 -0.0010771355
#> C04 C04  0.007357972 -0.0012901654
#> C05 C05  0.028412553  0.0005464276
#> C06 C06  0.066339390  0.0029356615

shape$plot

```



Input points must follow each boundary in order. The vignette reports reproducible scores without assigning a tumor phenotype.

#### 0.4 Volumetric ISPAT-3D input

```

volume <- demo_spatial_3d_data(n = 72, volumes = 2, seed = 2026)
volume_features <- c("epithelial", "APC", "CTL", "Treg", "T_helper")
volume_mapping <- list(
  x = "x", y = "y", z = "z", zone = "zone", id = "volume_id"
)

```

```

)
validate_dataset(volume, volume_mapping)[c("ok", "n", "p")]
#> $ok
#> [1] TRUE
#>
#> $n
#> [1] 72
#>
#> $p
#> [1] 10

```

The public ISPAT-3D engine is opt-in and is not executed during vignette builds.

```

remotes::install_github("sagnikbhadury/ISPAT-3D")

ispat3d <- run_analysis(
  "ispat3d", volume, volume_mapping, volume_features,
  list(kernel = "Matern", factor_fit = "CAVI", threshold = 0.15, cores = 1L)
)
ispat3d$summary
head(ispat3d$table)

```

The 72-row dataset is a software demonstration. Registered coordinates, section alignment, zones, and study-specific sampling must be established before using real volumetric data.