Package: thyroidBRS (via r-universe)

November 25, 2026

Title BRAF-RAS Score (BRS) Classifier for Papillary Thyroid Carcinoma

Version 1.0.0

Description Computes the BRAF-RAS Score (BRS), a 71-gene expression-based signature that quantifies whether a papillary thyroid carcinoma (PTC) sample resembles a BRAF-V600E-mutant or a RAS-mutant transcriptional profile, following Agrawal et al. (2014) <doi:10.1016/j.cell.2014.09.050>, Extended Experimental Procedures, section 14.1. It is a re-implementation from that description rather than the original code, which was never released: it reproduces the published classification (99.7 percent against driver-mutation status under cross-validation) but not the exact published score values. The original study only scored the subset of samples for which both exome and RNA sequencing data were available, since exome data was needed to define the two reference groups (but not to score additional samples). This package lets you fit the reference centroids from any cohort with known BRAF-mutant/RAS-mutant labels and then score new samples — including ones with expression data only — using just their gene expression. Expression can be supplied as a matrix, a SummarizedExperiment or an ExpressionSet.

License MIT + file LICENSE

Encoding UTF-8

biocViews Software, Classification, DriverMutation, GeneExpression, Transcriptomics, RNASeq, Microarray, Sequencing, BiomedicalInformatics

Roxygen list(markdown = TRUE)

Depends R (>= 4.5)

Imports SummarizedExperiment, methods, stats

Suggests Biobase, BiocStyle, knitr, rmarkdown, testthat (>= 3.1.5)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/camilasauria/thyroidBRS

BugReports https://github.com/camilasauria/thyroidBRS/issues

Config/roxygen2/version 8.1.0

RoxygenNote 7.3.3

Config/pak/sysreqs zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:57:05 UTC

RemoteUrl https://github.com/bioc/thyroidBRS

RemoteRef RELEASE_3_24

RemoteSha 4d5d2726659d6fb0d4391b4e76128094f11463e3

Contents

brs_fit 2
brs_genes 5
brs_score 6
containers 8
predict.brs_fit 9
validate_brs 11

Index **13**

`brs_fit` *Fit BRAF-RAS Score (BRS) reference centroids*

Description

Computes the two reference centroids (BRAF-mutant and RAS-mutant) needed to score the BRAF-RAS Score (BRS) of Agrawal et al. (2014), from a cohort where at least some samples have a known driver mutation. The original study needed exome sequencing to assign those labels, but that is only required once, to fit the centroids. Scoring additional samples afterwards (`predict.brs_fit()`) only needs their gene expression.

Usage

```
brs_fit(expr, labels, genes = NULL, log2_transform = FALSE, assay = NULL)

## S3 method for class 'brs_fit'
print(x, ...)
```

Arguments

expr	Gene expression, features (genes) as rows and samples as columns: a numeric matrix, a SummarizedExperiment::SummarizedExperiment or a Biobase::ExpressionSet (see containers). Row names must be gene symbols; column names must be sample identifiers. Expression should already be on a log-like scale (e.g. log2(TPM + 1)); see log2_transform if it isn't.
labels	A named character vector giving the reference group for the samples that have one, or a single string naming a column of the object's colData()/pData(). Names must match a subset of colnames(expr). Values are matched case-insensitively and after collapsing runs of non-alphanumeric characters, so "BRAF V600E", "BRAF-like" and "braf_like" are equivalent. The BRAF group is named by "BRAF_V600E", "BRAF", "BRAF-like", "BRAF-mutant", "V600E" or "BVL"; the RAS group by "RAS", "RAS-like", "RAS-mutant", "KRAS", "NRAS", "HRAS" or "RL". Samples with any other value, NA, or simply absent from labels are treated as unlabeled and excluded from centroid fitting, but can still be scored with predict.brs_fit(). An unrecognised value warns: a mis-specified label column otherwise looks exactly like a cohort that has unlabeled samples. Each reference group needs at least two samples; fewer is an error.
genes	Character vector of gene symbols to use as the signature. Defaults to the alias-resolved names in brs_genes (71 genes).
log2_transform	Logical. If TRUE, applies log2(expr + 1) to expr before fitting (use this if you are passing raw TPM/counts instead of already-transformed values). The choice is stored in the returned object and reused by predict.brs_fit(), so the two can no longer disagree. Default FALSE.
assay	Name or index of the assay to use when expr is a SummarizedExperiment. Ignored otherwise. Defaults to the first assay, with a warning when there is more than one. For GDC data the first assay is raw counts, so use assay = "tpm_unstranded" with log2_transform = TRUE, or a normalized assay of your own.
x	A brs_fit object, for print().
...	Ignored; present for consistency with base::print().

Value

An object of class "brs_fit", a list with:

genes_used Signature genes actually present in expr and used to fit the centroids.

genes_missing Requested signature genes not found in expr.

genes_zero_variance Genes dropped because they had zero variance across the reference samples (uninformative for distance).

genes_missing_values Genes dropped because they had a missing value in at least one reference sample. Together with genes_zero_variance these account for every requested gene that was found but not used.

gene_mean, gene_sd Per-gene mean/SD from the reference samples, used to z-score any sample later in predict.brs_fit().

centroid_braf, centroid_ras Mean z-score profile of the BRAF-mutant and RAS-mutant reference samples.

n_braf, n_ras Number of reference samples in each group.

reference_samples Names of the samples used to fit, so that `validate_brs()` can tell resubstitution from real validation.

log2_transform The scale the centroids were fitted on.

Which samples define the reference groups

In Agrawal et al. the two reference sets are defined by **driver mutation**: set B is the BRAF-V600E-mutant tumors and set R is the RAS-mutant tumors. They are *not* the BRAF-like/RAS-like classes that the BRS itself produces.

This distinction matters when working with the published TCGA-THCA tables. The column `BRAF_RAF_class` (also exposed by `TCGAbiolinks::TCGAquery_subtype()`) is exactly `sign(BRAF_RAF_score)`: it is the *output* of the classifier. Fitting centroids on it is circular: it pulls 43 tumors with other drivers and 1 RAS-mutant tumor into the "BRAF" centroid, and 68 other-driver tumors into the "RAS" centroid. Use the mutation column (`BRAFV600E_RAS`, restricted to `BRAF_V600E` and `RAS`) to fit, and keep `BRAF_RAF_class` / `BRAF_RAF_score` for validation. See `vignette("thyroidBRS")`.

References

Agrawal N, Akbani R, Aksoy BA, et al. "Integrated Genomic Characterization of Papillary Thyroid Carcinoma." Cell. 2014;159(3):676-690. doi:10.1016/j.cell.2014.09.050, Extended Experimental Procedures section 14.1.

See Also

`predict.brs_fit()` to score samples with a fitted object, `validate_brs()` to check concordance against known labels.

Examples

```
set.seed(1)
genes <- head(brs_genes$current_symbol, 10)
samples <- paste0("S", 1:20)
group <- rep(c("BRAF_V600E", "RAS"), each = 10)

# Give the two groups the difference the signature is meant to detect, so
# the example shows the classifier working rather than fitting noise.
shift <- ifelse(
    brs_genes$up_in[match(genes, brs_genes$current_symbol)] == "RAS", 1, -1
)
expr <- matrix(rnorm(length(genes) * length(samples)),
    nrow = length(genes),
    dimnames = list(genes, samples)
)
expr <- expr + outer(shift, ifelse(group == "BRAF_V600E", -0.5, 0.5))
labels <- setNames(group[1:16], samples[1:16])

fit <- brs_fit(expr, labels, genes = genes)
```

fit

---

brs_genes *BRAF-RAS Score (BRS) signature genes*

Description

The 71-gene expression signature used to compute the BRAF-RAS Score (BRS) for papillary thyroid carcinoma (PTC), as published in Agrawal et al. (2014), Figure S7A. No machine-readable gene list was released with the paper, so the symbols were transcribed from the published heatmap and then verified against it row by row. The provenance and the checks that were run are written up in `data-raw/brs_genes.md`, which is in the package repository rather than the installed package: https://github.com/camilasauria/thyroidBRS/blob/main/data-raw/brs_genes.md.

Usage

```
brs_genes
```

Format

A data frame with 71 rows and 4 columns:

original_symbol Gene symbol exactly as it appears in the 2014 publication's Figure S7A.

current_symbol Current HGNC-approved symbol. Identical to `original_symbol` for genes whose symbol has not changed. Four were resolved against `org.Hs.eg.db`; `FLJ23867`, a cDNA clone name, was resolved from NCBI Gene to `QSOX1`. See the Alias resolution section.

block Which of the two row-clusters of Figure S7A the gene belongs to: `1` for the 13-gene block, `2` for the 58-gene block. Each block is listed alphabetically in the figure, which is what makes the transcription checkable: no gene is out of order within its block.

up_in Which reference group the gene is higher in, `"RAS"` for block 1 and `"BRAF"` for block 2. Refitting the centroids on TCGA-THCA recovers this split exactly: all 13 block-1 genes have a higher RAS centroid and all 58 block-2 genes a higher BRAF centroid, with no exceptions. That is an independent check on the gene list: a mis-transcribed symbol would land in the wrong block.

Alias resolution

Five labels in the 2014 figure are not approved HGNC symbols today. Four are former symbols: `FAM176A` (now `EVA1A`), `PVRL4` (`NECTIN4`), `TM7SF4` (`DCSTAMP`) and `ARNTL` (`BMAL1`). Expression matrices built from current GENCODE/HGNC annotation use the new symbols, so matching on `original_symbol` silently loses those genes. Use `current_symbol`, which is what `brs_fit()` does by default.

`FLJ23867` is neither a symbol nor an alias in `org.Hs.eg.db`, HGNC or GENCODE. It is a cDNA clone name: NCBI Gene record 200058 ("uncharacterized protein FLJ23867", 1q25.2) was withdrawn and merged into 5768, `QSOX1` (quiescin sulfhydryl oxidase 1), whose record lists FLJ23867 among its clone names. The merge dates from 2018, so in 2014 FLJ23867 was a Gene ID of its own

and the published row may have measured a transcript at that locus rather than the whole of QSOX1. QSOX1 is therefore the closest current gene, not necessarily the same feature. In TCGA-THCA it is higher in BRAF-V600E than in RAS tumors and tracks the rest of block 2.

Resolution was checked against org.Hs.eg.db; re-run data-raw/check_symbols.R when the annotation is updated. data-raw/ is not shipped in the installed package; it lives in the package repository, https://github.com/camilasauria/thyroidBRS.

Source

Agrawal N, Akbani R, Aksoy BA, et al. "Integrated Genomic Characterization of Papillary Thyroid Carcinoma." Cell. 2014;159(3):676-690. doi:10.1016/j.cell.2014.09.050, Figure S7A ("mRNA expression of the 71 genes used to derive the BRAF-V600E-RAS score (BRS) across 391 PTC samples").

Examples

```
brs_genes
# Signature genes under current symbols:
brs_genes$current_symbol
# The two blocks of Figure S7A:
table(brs_genes$block)
```

---

brs_score *Fit and score in one step*

Description

Convenience wrapper equivalent to predict(brs_fit(expr, labels, ...), newdata), for the common case of scoring the same cohort you used to fit the centroids (e.g. to extend a partially-labeled cohort to full coverage).

Usage

```
brs_score(
  expr,
  labels,
  newdata = expr,
  genes = NULL,
  log2_transform = FALSE,
  standardize = c("reference", "cohort", "rank"),
  assay = NULL
)
```

Arguments

expr	Gene expression, features (genes) as rows and samples as columns: a numeric matrix, a SummarizedExperiment::SummarizedExperiment or a Biobase::ExpressionSet (see containers). Row names must be gene symbols; column names must be sample identifiers. Expression should already be on a log-like scale (e.g. log2(TPM + 1)); see log2_transform if it isn't.
labels	A named character vector giving the reference group for the samples that have one, or a single string naming a column of the object's colData()/pData(). Names must match a subset of colnames(expr). Values are matched case-insensitively and after collapsing runs of non-alphanumeric characters, so "BRAF V600E", "BRAF-like" and "braf_like" are equivalent. The BRAF group is named by "BRAF_V600E", "BRAF", "BRAF-like", "BRAF-mutant", "V600E" or "BVL"; the RAS group by "RAS", "RAS-like", "RAS-mutant", "KRAS", "NRAS", "HRAS" or "RL". Samples with any other value, NA, or simply absent from labels are treated as unlabeled and excluded from centroid fitting, but can still be scored with predict.brs_fit(). An unrecognised value warns: a mis-specified label column otherwise looks exactly like a cohort that has unlabeled samples. Each reference group needs at least two samples; fewer is an error.
newdata	Expression to score: a matrix, a SummarizedExperiment or an ExpressionSet, as for expr. Defaults to expr itself, so brs_score(expr, labels) scores every sample in expr, including the ones without a label. assay applies to whichever of expr and newdata is a SummarizedExperiment (to both, when both are).
genes	Character vector of gene symbols to use as the signature. Defaults to the alias-resolved names in brs_genes (71 genes).
log2_transform	Logical. If TRUE, applies log2(expr + 1) to expr before fitting (use this if you are passing raw TPM/counts instead of already-transformed values). The choice is stored in the returned object and reused by predict.brs_fit(), so the two can no longer disagree. Default FALSE.
standardize	Passed to predict.brs_fit().
assay	Name or index of the assay to use when expr is a SummarizedExperiment. Ignored otherwise. Defaults to the first assay, with a warning when there is more than one. For GDC data the first assay is raw counts, so use assay = "tpm_unstranded" with log2_transform = TRUE, or a normalized assay of your own.

Value

See predict.brs_fit().

Examples

```
set.seed(1)
genes <- head(brs_genes$current_symbol, 10)
samples <- paste0("S", 1:20)
group <- rep(c("BRAF_V600E", "RAS"), each = 10)

# Give the two groups the difference the signature is meant to detect, so
```

```
# the example shows the classifier working rather than fitting noise.
shift <- ifelse(
    brs_genes$up_in[match(genes, brs_genes$current_symbol)] == "RAS", 1, -1
)
expr <- matrix(rnorm(length(genes) * length(samples)),
    nrow = length(genes),
    dimnames = list(genes, samples)
)
expr <- expr + outer(shift, ifelse(group == "BRAF_V600E", -0.5, 0.5))
labels <- setNames(group[1:16], samples[1:16])

brs_score(expr, labels, genes = genes)
```

`containers` *Expression container support*

Description

`brs_fit()`, `predict()` and `brs_score()` accept expression as a plain numeric matrix, a SummarizedExperiment::SummarizedExperiment or a `Biobase::ExpressionSet`. Whichever you pass, the signature is matched on row names, so those must be gene symbols.

Value

This page documents the input containers the package accepts rather than a function of its own. Whichever container the expression arrives in, `brs_fit()` returns a `"brs_fit"` object and `predict.brs_fit()` returns a data frame.

Choosing an assay

A `SummarizedExperiment` usually carries several assays (for GDC RNA-seq: `unstranded`, `tpm_unstranded`, `fpkm_unstranded`, ...). Pass the one you want by name through `assay`; the first assay is used if you do not, which for GDC data is raw counts and is almost certainly not what you want. An `ExpressionSet` has a single `exprs()` matrix and ignores `assay`.

Labels from the object

When `labels` is a bare character string of length one it is looked up as a column of `colData()` (or `pData()`), so `brs_fit(se, "driver_mutation")` works directly on an object that carries its own sample annotation. Sample names come from `colnames()`. A *named* vector of length one is a label for one sample, not a column name.

Whichever column you name, it has to hold driver-mutation status. On TCGA-THCA the column giving the published BRAF-like/RAS-like class is the classifier's own output, and fitting on it is circular; see the warning in `brs_fit()`.

Microarray data

Microarray expression works as input: the signature is well represented on common arrays and its structure is preserved there. What does not carry over from one platform to another is the *zero threshold* that separates the two classes, because the per-gene means and SDs stored in a `brs_fit` are those of the platform it was fitted on. Fit centroids on the same platform you are scoring whenever you have driver-mutation labels for it. Scoring array data against RNA-seq centroids still ranks samples correctly, but the class call is not trustworthy; `predict()` warns when every sample lands on the same side, which is the usual symptom.

`predict.brs_fit` *Score samples with a fitted BRS model*

Description

Computes the BRAF-RAS Score (BRS) and the resulting class for every sample in `newdata`, using the centroids and per-gene mean/SD stored in `object` (from `brs_fit()`). `newdata` samples do not need to have been part of the reference cohort used to fit those centroids, and do not need any mutation/exome information.

Usage

```
## S3 method for class 'brs_fit'
predict(
  object,
  newdata,
  log2_transform = NULL,
  standardize = c("reference", "cohort", "rank"),
  assay = NULL,
  ...
)
```

Arguments

`object`	A `"brs_fit"` object, as returned by `brs_fit()`.
`newdata`	Gene expression to score, in any of the forms `expr` accepts (see containers) and with the same gene symbol row naming as the `expr` originally passed to `brs_fit()`. Genes in `object$genes_used` missing from `newdata` are an error, since the distance calculation would then be inconsistent with how the centroids were defined.
`log2_transform`	Logical, or `NULL` (the default) to reuse whatever was used when fitting `object`. The stored per-gene mean/SD are on that scale, so overriding it is almost always a mistake and warns.
`standardize`	How to put `newdata` on the scale the centroids live on. `"reference"` (the default) uses the per-gene mean and SD frozen from the fit. `"cohort"` re-derives them from `newdata` itself, and `"rank"` replaces each sample's expression by its

	rank among the signature genes (not the whole transcriptome) before standardizing. The last two need at least 3 samples. See the section below before changing it.
assay	Assay to use when newdata is a SummarizedExperiment; see brs_fit().
...	Not used. Present for S3 consistency with stats::predict(); anything passed here is ignored, with a warning, so that a misspelled argument name is not silently dropped.

Value

A data frame with one row per sample in colnames(newdata): sample, brs_score (normalized distance difference; negative = more BRAF-like, positive = more RAS-like), brs_scaled (the same score rescaled to $[-1, +1]$ across the scored samples, NA when it cannot be determined) and brs_class ("Braf-like" or "Ras-like"). A sample with a missing value in a signature gene gets NA in all three, with a warning.

Scoring data from another platform

The centroids carry the per-gene mean and SD of the cohort they were fitted on. That is what makes a sample's score its own: it does not move when you score it alongside a different set of samples. It is also what breaks when newdata comes from another platform, because those means and SDs no longer describe it.

Measured on GSE33630 (Affymetrix, 49 papillary carcinomas) against centroids fitted on TCGA RNA-seq: all three settings rank the tumors almost identically, at Spearman 0.97, 0.98 and 0.92 against the array's own signature axis, and they disagree completely on where zero falls. "reference" calls 49 of 49 BRAF-like; "cohort" calls 42 and 7.

"cohort" and "rank" fix the threshold across platforms by paying for it elsewhere: each sample's score then depends on which samples it is scored with. Scoring TCGA subsets of different composition against the same centroids, "reference" gets 97.5% to 99.5% right whatever the class mix, while "cohort" drops to 8.4% on a cohort that happens to be all RAS-like.

So: score on the platform you fitted on and keep the default. Crossing platforms, prefer fitting new centroids there if you have driver-mutation labels. Failing that, use these settings for the *ranking* and treat the class call as unreliable. predict() warns when every sample lands on the same side of zero, which is the usual symptom.

How the score is computed

Following Agrawal et al. (2014), Extended Experimental Procedures 14.1, the score of a tumor t is the difference between its distance to the two centroids,

$$BRS(t) = \|v(t) - c(B)\|_2 - \|v(t) - c(R)\|_2$$

where the distance is a normalized Euclidean distance. Here v(t) is the sample's expression vector z-scored gene-by-gene against the reference cohort, and the norm is divided by the square root of the number of signature genes, so that scores stay comparable when the gene set shrinks (for example because some signature genes are absent from newdata).

Negative scores are BRAF-V600E-like and positive scores RAS-like. A score of exactly `0` is not defined by the paper; it is assigned to `"Ras-like"` here, consistent with treating the non-negative half as the RAS side.

The BRS values published for TCGA-THCA are additionally rescaled to $[-1, +1]$, by dividing the negative scores by the most negative value and the positive scores by the largest one. `brs_scaled` reproduces that rescaling. Note that it is a property of the *set of samples being scored*, not of a sample on its own: scoring a different cohort, or a single sample, gives a different (or undefined) rescaling. Use `brs_score` when you need a value that is stable per sample, and `brs_scaled` when you need a value on the same axis as the published figures.

Examples

```
set.seed(1)
genes <- head(brs_genes$current_symbol, 10)
samples <- paste0("S", 1:20)
group <- rep(c("BRAF_V600E", "RAS"), each = 10)

# Give the two groups the difference the signature is meant to detect, so
# the example shows the classifier working rather than fitting noise.
shift <- ifelse(
    brs_genes$up_in[match(genes, brs_genes$current_symbol)] == "RAS", 1, -1
)
expr <- matrix(rnorm(length(genes) * length(samples)),
    nrow = length(genes),
    dimnames = list(genes, samples)
)
expr <- expr + outer(shift, ifelse(group == "BRAF_V600E", -0.5, 0.5))
labels <- setNames(group[1:16], samples[1:16])

fit <- brs_fit(expr, labels, genes = genes)
predict(fit, expr[, 17:20])
```

`validate_brs` *Check BRS predictions against known labels*

Description

Compares `predict.brs_fit()` output against a reference label vector.

Usage

```
validate_brs(predictions, labels, fit = NULL)
```

Arguments

`predictions`	A data frame as returned by `predict.brs_fit()` / `brs_score()`.
`labels`	A named character vector, as in `brs_fit()`. Only samples present in both `predictions$sample` and `names(labels)` are compared.

`fit` Optionally the `"brs_fit"` used to produce `predictions`, so the overlap with its reference samples can be reported.

Value

A list with `n`, `n_concordant`, `pct_concordant`, `n_resubstituted` (samples that also defined the centroids, `NA` if `fit` was not given) and the confusion `table` (published vs. predicted). The table always carries both classes on both margins, even when only one of them occurs, so that indexing it by name is safe. Samples whose label names neither reference group are dropped with a warning and are not counted in `n`.

Resubstitution is not validation

Passing the same `labels` that were used to fit the centroids measures how well a nearest-centroid rule reproduces the labels that *defined* those centroids. That statistic is strongly optimistic: on 71 genes and 391 samples split 272/119, pure noise with no signal at all still resubstitutes at about 69%. Pass `fit` and this function will tell you how much of the comparison is resubstitution, and warn when all of it is.

For a real check, compare against labels that did not enter the fit. For TCGA-THCA, fit on the mutation groups and validate against the published `BRAF_RAF_class` and `BRAF_RAF_score`.

Examples

```
set.seed(1)
genes <- head(brs_genes$current_symbol, 10)
samples <- paste0("S", 1:20)
group <- rep(c("BRAF_V600E", "RAS"), each = 10)

# Give the two groups the difference the signature is meant to detect, so
# the example shows the classifier working rather than fitting noise.
shift <- ifelse(
    brs_genes$up_in[match(genes, brs_genes$current_symbol)] == "RAS", 1, -1
)
expr <- matrix(rnorm(length(genes) * length(samples)),
    nrow = length(genes),
    dimnames = list(genes, samples)
)
expr <- expr + outer(shift, ifelse(group == "BRAF_V600E", -0.5, 0.5))
labels <- setNames(group[1:16], samples[1:16])

# Fit on 12 samples and compare against all 20, so that 8 of the
# samples scored did not help define the centroids. Passing `fit` makes
# validate_brs() report how much of the comparison is resubstitution.
in_fit <- samples[c(1:6, 11:16)]
fit <- brs_fit(expr, labels[in_fit], genes = genes)
preds <- predict(fit, expr)
validate_brs(preds, setNames(group, samples), fit = fit)
```

Index

∗ **datasets**
brs_genes, 5

base::print(), *3*
brs_fit, 2
brs_fit(), *5*, *8–11*
brs_genes, *3*, 5, *7*
brs_score, 6
brs_score(), *11*

containers, *3*, *7*, 8, *9*

predict.brs_fit, 9
predict.brs_fit(), *2–4*, *7*, *8*, *11*
print.brs_fit *(*brs_fit*)*, 2

stats::predict(), *10*
SummarizedExperiment::SummarizedExperiment, *3*, *7*, *8*

validate_brs, 11
validate_brs(), *4*