Package 'RiskyCNV'

June 5, 2026

Title Risk Analysis of Genomic Copy Number Variation

Version 0.1.0

Description Provides a complete seven-step workflow for copy number variation (CNV) analysis applicable to any disease or condition where samples with genomic copy number data is available. Supports built-in grading and risk stratification presets for seven major cancers (viz. prostate, breast, colorectal, lung, cervical, lymphoma, melanoma) based on clinically validated systems including ISUP Grade Groups, Nottingham Grading System, Dukes staging, IASLC TNM, FIGO, Ann Arbor/Lugano classification, and Breslow depth. Generalizable to other disease types. An automatic mode derives a normalised Risk Score from the data using min-max normalisation and adaptive binning. Custom user-defined thresholds are supported for any other disease type. Downstream functions for CNV aberration detection, recurrence analysis, gene annotation, CNV matrix generation, and CNV-RNA expression correlation are disease-type agnostic.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr, GenomicRanges, rlang, S4Vectors, stats, tidyr, tools, utils

Suggests BiocManager, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

biocViews CopyNumberVariation, GenomicVariation, RNASeq

Config/testthat/edition 3

NeedsCompilation no

Author Ashok Palaniappan [aut, cre] (ORCID: <https://orcid.org/0000-0003-2841-9527>), Priyanka Ramesh [aut], Ida Titus [aut], Sangeetha Muthamilselvan [aut]

Maintainer Ashok Palaniappan <apalania@scbt.sastra.edu>

Repository CRAN

Date/Publication 2026-06-05 15:00:17 UTC

Contents

`aberration` *Detect Copy Number Aberrations (Gains and Losses)*

Description

Reads a CNV (Copy Number Variation) data file and identifies genomic segments showing significant aberrations (gains or losses) based on a user-defined effect size threshold. Results are split by chromosome and returned as a named list.

Usage

```
aberration(cnv_data_file, effect_size = 0.3)
```

Arguments

`cnv_data_file`	Character. Path to the CNV data file (whitespace-delimited, with a header). Must contain columns: `Chromosome`, `Start`, `End`, `Num_Probes`, `Segment_Mean`, and `Sample`.
`effect_size`	Numeric. Threshold for calling aberrations. Segments with `Segment_Mean` > `effect_size` are called Gains; segments with `Segment_Mean` < `-effect_size` are called Losses. Default is `0.3`.

Details

Segments with `Segment_Mean` between `-effect_size` and `effect_size` (inclusive) are considered neutral and excluded from the output. The default threshold of 0.3 is widely used in TCGA-based CNV analyses. This function is cancer-type agnostic and can be applied to CNV data from any solid tumour.

Value

A named list where each element corresponds to a chromosome (e.g., "1", "2", ...) and contains a data frame of aberrant segments for that chromosome. Each data frame includes the columns: Chromosome, Start, End, Num_Probes, Segment_Mean, Sample, Aberration (Gain or Loss), and Aberration_Code (1 = Gain, 0 = Loss).

References

Mermel CH, et al. (2011). GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome Biol*, 12(4):R41.

Examples

```
cnv_file <- system.file("extdata", "cnv_data.txt", package = "RiskyCNV")
aberrations <- aberration(
  cnv_data_file = cnv_file,
  effect_size   = 0.3
)
print(aberrations[["2"]])
```

---

annotate *Annotate CNV Regions with Gene Symbols*

Description

Finds the overlap between a gene annotation file and a recurrent CNV file using genomic ranges, and annotates each CNV region with the corresponding gene symbol. Requires the **GenomicRanges** package.

Usage

```
annotate(
  genes_file,
  risk_file,
  output_dir = ".",
  seqnames_field_genes = "Chr",
  start_field_genes = "Start",
  end_field_genes = "End",
  gene_symbol_field = "GeneSymbol",
  seqnames_field_risk = "Chr",
  start_field_risk = "Start",
  end_field_risk = "End",
  sample_field = "Sample",
  segment_mean_field = "Segment_Mean"
)
```

Arguments

genes_file	Character. Path to the gene annotation CSV file. Must contain chromosome, start, end, and gene symbol columns (see parameters below for defaults).
risk_file	Character. Path to the recurrent CNV CSV file (e.g., the file path returned by recurrent). Must contain sample, chromosome, start, end, and segment mean columns.
output_dir	Character. Directory where the annotated CSV will be saved. Default is the current directory ("..").
seqnames_field_genes	Character. Column name for chromosome in the gene file. Default is "Chr".
start_field_genes	Character. Column name for start position in the gene file. Default is "Start".
end_field_genes	Character. Column name for end position in the gene file. Default is "End".
gene_symbol_field	Character. Column name for gene symbols in the gene file. Default is "GeneSymbol".
seqnames_field_risk	Character. Column name for chromosome in the CNV file. Default is "Chr".
start_field_risk	Character. Column name for start position in the CNV file. Default is "Start".
end_field_risk	Character. Column name for end position in the CNV file. Default is "End".
sample_field	Character. Column name for sample IDs in the CNV file. Default is "Sample".
segment_mean_field	Character. Column name for segment mean values in the CNV file. Default is "Segment_Mean".

Details

This function uses GenomicRanges::findOverlaps with type = "within" to find genes that fall entirely within each CNV region. This function is cancer-type agnostic and can be applied to CNV data from any solid tumour with a compatible gene annotation reference file.

Value

A data frame containing annotated CNV regions with columns: Sample, GeneSymbol, Segment_Mean, Chr, Start, End. The result is also written to a timestamped CSV file in output_dir.

Examples

```
genes_file <- system.file("extdata", "gene_annotation.csv",
                          package = "RiskyCNV")
cnv_file   <- system.file("extdata", "annotated_cnv.csv",
                          package = "RiskyCNV")
annotated  <- annotate(
  genes_file = genes_file,
  risk_file  = cnv_file,
  output_dir = tempdir()
```

```
)
head(annotated)
```

`classify_risk` *Classify Samples into Risk Categories*

Description

Reads a CSV file containing sample metadata and assigns each sample to a risk category based on a specified scoring column. Supports built-in presets for seven major disease types, fully custom user-defined risk boundaries, or automatic classification using a normalised Risk Score derived from the data itself.

Usage

```
classify_risk(
  file_path,
  column_name,
  disease_type = "auto",
  n_groups = 3,
  score_min = NULL,
  score_max = NULL,
  risk_groups = NULL,
  output_dir = NULL
)
```

Arguments

`file_path`	Character. Path to the input CSV file containing sample metadata.
`column_name`	Character. Name of the column containing the grading or staging score (e.g., Gleason score, Nottingham score, TNM stage).
`disease_type`	Character. Disease type for built-in preset risk groupings. Supported values: `"prostate"`, `"breast"`, `"colorectal"`, `"lung"`, `"cervical"`, `"lymphoma"`, `"melanoma"`. Use `"custom"` to supply your own groupings via the `risk_groups` argument. Use `"auto"` to automatically classify using a normalised Risk Score derived from the data. Default is `"auto"`.
`n_groups`	Integer. Number of risk groups to create. Only used when `disease_type = "auto"`. Must be between 2 and 5. Default is `3`.
	2 groups low_risk, high_risk
	3 groups low_risk, intermediate_risk, high_risk
	4 groups very_low_risk, low_risk, high_risk, very_high_risk
	5 groups very_low_risk, low_risk, intermediate_risk, high_risk, very_high_risk
`score_min`	Numeric or NULL. Minimum possible value of the score. If NULL (default), automatically detected from the data.

score_max	Numeric or NULL. Maximum possible value of the score. If NULL (default), automatically detected from the data.
risk_groups	Named list of functions. Required only when disease_type = "custom". Each element must be a function that takes a numeric or character vector and returns a logical vector. The name of each element becomes the risk group label.
output_dir	Character or NULL. Directory to save the output CSV file. If NULL (default), output is saved in the same directory as the input file.

Details

When disease_type = "auto", the function computes a normalised Risk Score for each sample using min-max normalisation:

$$RiskScore = \frac{score - min(score)}{max(score) - min(score)}$$

The Risk Score ranges from 0 (lowest risk) to 1 (highest risk). Risk group boundaries are then determined automatically:

- If the score distribution is approximately symmetric (skewness between -0.5 and +0.5), equal-width boundaries are used, dividing the 0-1 range into n_groups equal intervals.
- If the score distribution is skewed (skewness outside -0.5 to +0.5), quantile-based boundaries are used, ensuring approximately equal numbers of samples per group.

The splitting method chosen is reported via a message. Risk group labels are generated automatically based on n_groups.

Built-in presets use clinically validated risk stratification systems:

prostate D'Amico classification (D'Amico et al., 1998): low_risk (<=6), intermediate_risk (7), high_risk (>=8).

breast Nottingham Prognostic Index (Galea et al., 1992): low_risk (3-5), intermediate_risk (6-7), high_risk (8-9).

colorectal Dukes-based risk (Dukes, 1932): low_risk (A), intermediate_risk (B/C), high_risk (D).

lung TNM stage-based (Goldstraw et al., 2016): low_risk (I), intermediate_risk (II/III), high_risk (IV).

cervical FIGO stage-based (Bhatla et al., 2019): low_risk (I), intermediate_risk (II/III), high_risk (IV).

lymphoma Ann Arbor/Lugano (Cheson et al., 2014): limited (I/II), advanced (III/IV).

melanoma Breslow depth (Breslow, 1970): low_risk (<=1.0mm), intermediate_risk (1.0-4.0mm), high_risk (>4.0mm).

Value

A named list where each element corresponds to a risk group and contains the sample IDs belonging to that group. The number of elements matches the number of risk groups detected or specified.

References

- D'Amico AV, et al. (1998). Biochemical outcome after radical prostatectomy. *JAMA*, 280(11):969-974.
- Galea MH, et al. (1992). The Nottingham prognostic index. *Breast Cancer Res Treat*, 22(3):207-219.
- Dukes CE. (1932). The classification of cancer of the rectum. *J Pathol Bacteriol*, 35:323-332.
- Goldstraw P, et al. (2016). The IASLC Lung Cancer Staging Project. *J Thorac Oncol*, 11(1):39-51.
- Bhatla N, et al. (2019). Revised FIGO staging for carcinoma of the cervix uteri. *Int J Gynaecol Obstet*, 145(1):129-135.
- Cheson BD, et al. (2014). The Lugano Classification. *J Clin Oncol*, 32(27):3059-3068.
- Breslow A. (1970). Thickness and depth of invasion in the prognosis of cutaneous melanoma. *Ann Surg*, 172(5):902-908.

Examples

```
# Auto mode - let the function decide risk grouping (any disease)
sample_file <- system.file("extdata", "sample_data.csv",
                           package = "RiskyCNV")
result <- classify_risk(
  file_path    = sample_file,
  column_name  = "gleason_score",
  disease_type = "auto",
  n_groups     = 3,
  output_dir   = tempdir()
)
print(names(result))

# Prostate cancer preset
result_prostate <- classify_risk(
  file_path    = sample_file,
  column_name  = "gleason_score",
  disease_type = "prostate",
  output_dir   = tempdir()
)
print(result_prostate$low_risk)

# Custom risk groups for any disease

result_custom <- classify_risk(
  file_path    = "samples.csv",
  column_name  = "risk_score",
  disease_type = "custom",
  risk_groups  = list(
    "low_risk"  = function(x) x <= 5,
    "high_risk" = function(x) x > 5
  ),
  output_dir   = tempdir()
)
```

`correlate_with_expr` *Correlate CNV Profiles with Gene Expression Data*

Description

Computes Pearson correlations between CNV segment means and RNA expression values for each gene present in both datasets. RNA data is log2-transformed prior to analysis. Three result files are written: all correlations, those with p-value < 0.05, and those with both p-value < 0.05 and correlation coefficient > 0.8.

Usage

```
correlate_with_expr(cnv_file, rna_file)
```

Arguments

`cnv_file`	Character. Path to the CNV matrix CSV file (output of `create_CNVMatrix`). Rows are samples; first column is sample IDs; remaining columns are gene symbols.
`rna_file`	Character. Path to the RNA expression CSV file. Rows are genes; first column is gene names; remaining columns are sample IDs (trimmed to 12 characters for TCGA-style matching).

Details

Sample IDs in the RNA file are trimmed to 12 characters to match TCGA-style identifiers. Infinite values from `log2(0)` are replaced with 0. Pearson correlation is computed using `stats::cor.test` with `use = "complete.obs"`. This function is cancer-type agnostic.

Value

A named list with three data frames:

all_correlations All computed Pearson correlations with columns `gene`, `cor_val`, `p.value`.

significant Subset where `p.value` < `0.05`.

high_correlation Subset where `p.value` < `0.05` AND `cor_val` > `0.8`.

Results are also written to CSV files in the temporary directory.

References

Chin L, et al. (2011). Making sense of cancer genomic data. *Genes Dev*, 25(6):534-555.

Examples

```
cnv_file <- system.file("extdata", "cnv_matrix.csv", package = "RiskyCNV")
rna_file <- system.file("extdata", "rna_data.csv",   package = "RiskyCNV")
results  <- correlate_with_expr(
  cnv_file = cnv_file,
  rna_file = rna_file
)
head(results$all_correlations)
```

`create_CNVMatrix` *Create a CNV Expression Matrix*

Description

Takes an annotated CNV CSV file (output of `annotate`) and reshapes it into a wide-format matrix where rows are samples, columns are gene symbols, and values are mean segment means. Duplicate sample-gene combinations are resolved by taking the mean.

Usage

```
create_CNVMatrix(input_file)
```

Arguments

`input_file`	Character. Path to the input CSV file containing columns `Sample`, `GeneSymbol`, and `Segment_Mean`.

Details

Duplicate `Sample`-`GeneSymbol` combinations are summarised by taking their mean `Segment_Mean` before pivoting, avoiding list-column issues in the output. This function is cancer-type agnostic.

Value

A data frame in wide format with samples as rows and gene symbols as columns. Missing values are represented as `NA`. The matrix is also saved as a timestamped CSV file in the temporary directory.

Examples

```
annot_file <- system.file("extdata", "annotated_cnv.csv",
                          package = "RiskyCNV")
cnv_mat <- create_CNVMatrix(annot_file)
dim(cnv_mat)
head(cnv_mat)
```

`extract_metadata` *Extract Sample Metadata and Classify into Grade or Stage Groups*

Description

Reads a CSV file containing sample metadata and classifies each sample into grade or stage groups based on a specified scoring column. Supports built-in presets for seven major disease types, fully custom user-defined thresholds, or automatic classification using a normalised Risk Score derived from the data itself.

Usage

```
extract_metadata(
  file_path,
  column_name,
  disease_type = "auto",
  pattern_col = NULL,
  n_groups = 3,
  group_type = "grade",
  score_min = NULL,
  score_max = NULL,
  thresholds = NULL,
  output_dir = NULL
)
```

Arguments

`file_path`	Character. Path to the input CSV file containing sample metadata.
`column_name`	Character. Name of the column containing the grading or staging score (e.g., Gleason score, Nottingham score, TNM stage).
`disease_type`	Character. Disease type for built-in preset thresholds. Supported values: `"prostate"`, `"breast"`, `"colorectal"`, `"lung"`, `"cervical"`, `"lymphoma"`, `"melanoma"`. Use `"custom"` to supply your own thresholds via the `thresholds` argument. Use `"auto"` to automatically classify using a normalised Risk Score derived from the data. Default is `"auto"`.
`pattern_col`	Character or NULL. Only used when `disease_type = "prostate"`. Name of the column containing the primary Gleason pattern (pattern1). When provided, Grade Group 2 (Gleason 3+4=7, primary pattern 3) and Grade Group 3 (Gleason 4+3=7, primary pattern 4) are distinguished accurately. If NULL (default), all Gleason 7 samples are assigned to Grade Group 2.
`n_groups`	Integer. Number of grade or stage groups to create. Only used when `disease_type = "auto"`. Must be between 2 and 5. Default is `3`.
`group_type`	Character. Type of group labels to generate. Only used when `disease_type = "auto"`. One of `"grade"`, `"stage"`, or `"risk"`. Default is `"grade"`.
	grade Labels as Grade 1, Grade 2, ...

stage Labels as Stage I, Stage II, ...

risk Labels as low_risk, intermediate_risk, high_risk, ...

`score_min` Numeric or NULL. Minimum possible value of the score. If NULL (default), automatically detected from the data.

`score_max` Numeric or NULL. Maximum possible value of the score. If NULL (default), automatically detected from the data.

`thresholds` Named list of functions. Required only when `disease_type = "custom"`. Each element must be a function that takes a numeric or character vector and returns a logical vector. The name of each element becomes the grade or stage group label.

`output_dir` Character or NULL. Directory to save the output CSV file. If NULL (default), output is saved in the same directory as the input file.

Details

For prostate cancer, an optional `pattern_col` parameter allows accurate distinction between Grade Group 2 (Gleason 3+4=7) and Grade Group 3 (Gleason 4+3=7) using the primary histological pattern column.

Prostate cancer Grade Group 2 vs Grade Group 3 distinction:

Both Grade Group 2 (Gleason 3+4=7) and Grade Group 3 (Gleason 4+3=7) have the same total Gleason score of 7, making them indistinguishable from the total score alone. The primary histological pattern determines the correct assignment:

- Primary pattern 3 + secondary pattern 4 $\rightarrow$ Grade Group 2
- Primary pattern 4 + secondary pattern 3 $\rightarrow$ Grade Group 3

Supply the name of the primary pattern column via `pattern_col` (typically `"pattern1"`) to enable this distinction. If `pattern_col` is not supplied, all Gleason 7 samples are assigned to Grade Group 2 and a message is shown.

Auto mode:

When `disease_type = "auto"`, the function computes a normalised Risk Score for each sample using min-max normalisation:

$$RiskScore = \frac{score - min(score)}{max(score) - min(score)}$$

The Risk Score ranges from 0 (lowest risk) to 1 (highest risk). Group boundaries are determined automatically based on distribution skewness:

- Symmetric distribution (skewness between -0.5 and +0.5): equal-width boundaries
- Skewed distribution (skewness outside -0.5 to +0.5): quantile-based boundaries

Value

A named list where each element corresponds to a grade or stage group and contains the sample IDs belonging to that group.

References

- Epstein JI, et al. (2016). The 2014 ISUP Consensus Conference on Gleason Grading. *Am J Surg Pathol*, 40(2):244-252.
- Elston CW & Ellis IO. (1991). Pathological prognostic factors in breast cancer. *Histopathology*, 19(5):403-410.
- Dukes CE. (1932). The classification of cancer of the rectum. *J Pathol Bacteriol*, 35:323-332.
- Goldstraw P, et al. (2016). The IASLC Lung Cancer Staging Project. *J Thorac Oncol*, 11(1):39-51.
- Bhatla N, et al. (2019). Revised FIGO staging for carcinoma of the cervix uteri. *Int J Gynaecol Obstet*, 145(1):129-135.
- Cheson BD, et al. (2014). The Lugano Classification. *J Clin Oncol*, 32(27):3059-3068.
- Breslow A. (1970). Thickness and depth of invasion in the prognosis of cutaneous melanoma. *Ann Surg*, 172(5):902-908.

Examples

```
sample_file <- system.file("extdata", "sample_data.csv",
                           package = "RiskyCNV")

# Prostate preset — without pattern column (Grade Group 2 and 3 merged)
result <- extract_metadata(
  file_path    = sample_file,
  column_name  = "gleason_score",
  disease_type = "prostate",
  output_dir   = tempdir()
)
print(names(result))

# Prostate preset — with pattern column (Grade Group 2 and 3 distinguished)
result_full <- extract_metadata(
  file_path    = sample_file,
  column_name  = "gleason_score",
  disease_type = "prostate",
  pattern_col  = "pattern1",
  output_dir   = tempdir()
)
print(names(result_full))

# Auto mode
result_auto <- extract_metadata(
  file_path    = sample_file,
  column_name  = "gleason_score",
  disease_type = "auto",
  n_groups     = 3,
  group_type   = "grade",
  output_dir   = tempdir()
)
print(names(result_auto))
```

```
# Custom thresholds
result_custom <- extract_metadata(
  file_path    = sample_file,
  column_name  = "gleason_score",
  disease_type = "custom",
  thresholds   = list(
    "Stage I"   = function(x) x <= 6,
    "Stage II"  = function(x) x == 7,
    "Stage III" = function(x) x == 8,
    "Stage IV"  = function(x) x > 8
  ),
  output_dir   = tempdir()
)
print(names(result_custom))
```

`recurrent` *Identify Recurrent Copy Number Variations by Risk Group*

Description

Filters a CNV data file for samples belonging to a specified risk group and identifies genomic regions that recur across multiple samples above a given threshold. Results are saved as a CSV file.

Usage

```
recurrent(x, risk_level, cnv_data_file, threshold = 2)
```

Arguments

`x`	A named list of sample ID vectors, as returned by `classify_risk`. Each element name corresponds to a risk group label (e.g., `"low_risk"`, `"intermediate_risk"`, `"high_risk"`).
`risk_level`	Character. The risk group to analyse. Must be a name present in `x`.
`cnv_data_file`	Character. Path to the CNV data file (whitespace-delimited, with a header). Must contain columns: `Sample`, `Chromosome`, `Start`, `End`, `Num_Probes`, `Segment_Mean`.
`threshold`	Numeric. Minimum number of samples a CNV region must appear in to be considered recurrent. Default is `2`.

Details

Sample IDs in the CNV file are trimmed to 12 characters and hyphens are replaced with dots to match standard TCGA-style identifiers. The output CSV is saved inside a timestamped subdirectory under `recurrent_cnv/` in the temporary directory. This function is cancer-type agnostic.

Value

Character. The file path of the saved CSV file containing the recurrent CNV regions for the specified risk group.

Examples

```
sample_file <- system.file("extdata", "sample_data.csv", package = "RiskyCNV")
cnv_file    <- system.file("extdata", "cnv_data.txt",    package = "RiskyCNV")
risk_result <- classify_risk(
  file_path    = sample_file,
  column_name  = "gleason_score",
  disease_type = "prostate",
  output_dir   = tempdir()
)
output_path <- recurrent(
  x             = risk_result,
  risk_level    = "low_risk",
  cnv_data_file = cnv_file,
  threshold     = 2
)
print(output_path)
```

RiskyCNV *RiskyCNV: A General-Purpose CNV Analysis Workflow for Disease Risk Stratification*

Description

Provides a complete seven-step workflow for copy number variation (CNV) analysis applicable to any disease or condition where genomic copy number data is available. The package supports three classification approaches: built-in grading and risk stratification presets for seven major disease types (prostate, breast, colorectal, lung, cervical, lymphoma, melanoma) based on clinically validated scoring systems; an automatic mode that derives a normalised Risk Score from the data itself using min-max normalisation and adaptive binning; and fully user-defined custom thresholds for any disease type not covered by the presets. Downstream functions for CNV aberration detection, recurrence analysis, gene annotation, CNV matrix generation, and CNV-RNA expression correlation are disease-type agnostic and work with any genomic dataset.

Workflow

The recommended analysis pipeline proceeds in seven steps:

1. `extract_metadata` — Classify samples into grade or stage groups based on a clinical scoring parameter. Supports disease-specific presets, auto mode, and custom thresholds.
2. `classify_risk` — Stratify samples into risk categories. Supports disease-specific presets, auto mode, and custom thresholds.
3. `aberration` — Detect CNV gains and losses from segmented CNV data using a user-defined effect size threshold.
4. `recurrent` — Identify CNV regions that recur across multiple samples within a risk group.
5. `annotate` — Annotate recurrent CNV regions with gene symbols using genomic range overlaps.

6. `create_CNVMatrix` — Construct a sample x gene CNV expression matrix.
7. `correlate_with_expr` — Compute Pearson correlations between CNV profiles and RNA expression data.

Classification Modes

Disease presets Built-in clinically validated thresholds for prostate (ISUP/Gleason), breast (Nottingham), colorectal (Dukes), lung (IASLC/TNM), cervical (FIGO), lymphoma (Ann Arbor/Lugano), and melanoma (Breslow depth).

Auto mode Automatically computes a normalised Risk Score using min-max normalisation. Group boundaries are determined adaptively based on distribution skewness — equal-width for symmetric distributions, quantile-based for skewed distributions. Works for any numeric scoring column without prior knowledge of the scoring system.

Custom mode Users supply their own threshold functions for any disease type, scoring system, or number of groups.

Supported Disease Types (Built-in Presets)

prostate (ISUP/Gleason), breast (Nottingham/NGS), colorectal (Dukes), lung (IASLC/TNM), cervical (FIGO), lymphoma (Ann Arbor/Lugano), and melanoma (Breslow depth).

Author(s)

Maintainer: Ashok Palaniappan <apalania@scbt.sastra.edu> (ORCID)

Authors:

- Priyanka Ramesh
- Ida Titus
- Sangeetha Muthamilselvan

References

- Epstein JI, et al. (2016). *Am J Surg Pathol*, 40(2):244-252.
- Elston CW & Ellis IO. (1991). *Histopathology*, 19(5):403-410.
- Dukes CE. (1932). *J Pathol Bacteriol*, 35:323-332.
- Goldstraw P, et al. (2016). *J Thorac Oncol*, 11(1):39-51.
- Bhatla N, et al. (2019). *Int J Gynaecol Obstet*, 145(1):129-135.
- Cheson BD, et al. (2014). *J Clin Oncol*, 32(27):3059-3068.
- Breslow A. (1970). *Ann Surg*, 172(5):902-908.

Index