# diffSeqPatterns

Chloe H. Lee
University of Oxford, UK

2022-01-18

The diffSeqPatterns is an R package to comprehensively analyse differential sequence patterns between any two groups of peptides. The diffSeqPatterns provides functionality to analyse and visualize differential patterns on:

- **Position-specific amino acid usage**: Enrichment or depletion of amino acid usage in each positions of peptides having same lengths. Enrichment scores can be visualized by heatmap or sequence logo.
- **N-grams**: An n-gram is a contiguous sequence of n-amino acids, often used to discover motifs in biological sequences that are functionally related. Top differential n-grams can be visualized by scatter or bar plot.
- **Position-specific k-mer motifs**: A position-specific k-mer is contiguous or non-contiguous sequence of k amino acids, where e.g. .M.W. denotes MW pattern at P2 and P4 of 5 amino acid peptides (restricted to peptides of same lengths). Top position-specific k-mer motifs are visualized by scatter or bar plot.
- **Inter-sequence distance or alignment score**: Pairwise distance (e.g. Hamming, levenshtein, Jaccard distance) or global/local alignment using substitution matrix of interest (e.g. BLOSUM62) to analyse sequence homology or evolutionary relationship between peptides. Clusters of peptides sharing high sequence homology or evolutionary relationship can be visualized by heatmap or network graph.

We believe analysis and visualization tools in diffSeqPatterns will facilitate identification of conserved patterns in biologically active peptides, such as cancer neoepitopes, SARS-CoV-2 epitopes or antimicrobial peptides.

## Installing diffSeqPattern package

To install package from CRAN:

```
library(diffSeqPatterns)
```

To install the latest version of diffSeqPatterns package:

```
#install the development version from GitHub:
install.packages('devtools')
devtools::install_github('ChloeHJ/diffSeqPatterns', build_vignettes = TRUE)
```

## Data

The input data are two lists of peptide sequences i.e. Positive peptides to analyse for enrichment and Negative peptides to analyse for depletion. The Negative data can be either experimentally validated 'negative' peptides or a list of randomly generated peptides to serve as a background.

We demonstrated the diffSeqPatterns on DMF5 T cell antigens to investigate sequence patterns associated with T cell immune response. Gee et al. used yeast-display peptide-HLA-A*02:01 library to screen for antigens against DMF5 T cells [1]. We retrieved sequences identified from round 3 deep-sequencing of the DMF5 10mer library (61 unique peptides) and analysed for enriched sequence patterns compared to 200 randomly sampled peptides from 10mer library ('Background') [2]. As previous studies showed contact positions i.e. P3-P9 of 10aa peptides are associated with T cell recognition3, we analysed sequence patterns in P3-P9 [3].

- `DMF5_pos_peptides`: 55 unique peptides recognized by DMF5 T cells at their contact positions i.e. P3-P9 of 10 amino acid peptide
- `DMF5_neg_peptides`: 200 randomly sampled non-epitopes at contact positions

```
library(diffSeqPatterns)
data('DMF5_pos_peptides', 'DMF5_neg_peptides', 'DMF5_antigen_table',  package = 'diffSeqPatterns')
```

## Position-specific amino acid usage

We generate probability frequency of each amino acids in each position [4]. doi:10.18129/B9.bioc.Biostrings.] using position specific scoring matrix (PSSM), standardize PSSMs by centre and scaling, and compute difference in standardised PSSMs between Positive and Negative peptides. Due to position specificity, analysis is restricted to peptides having same lengths.

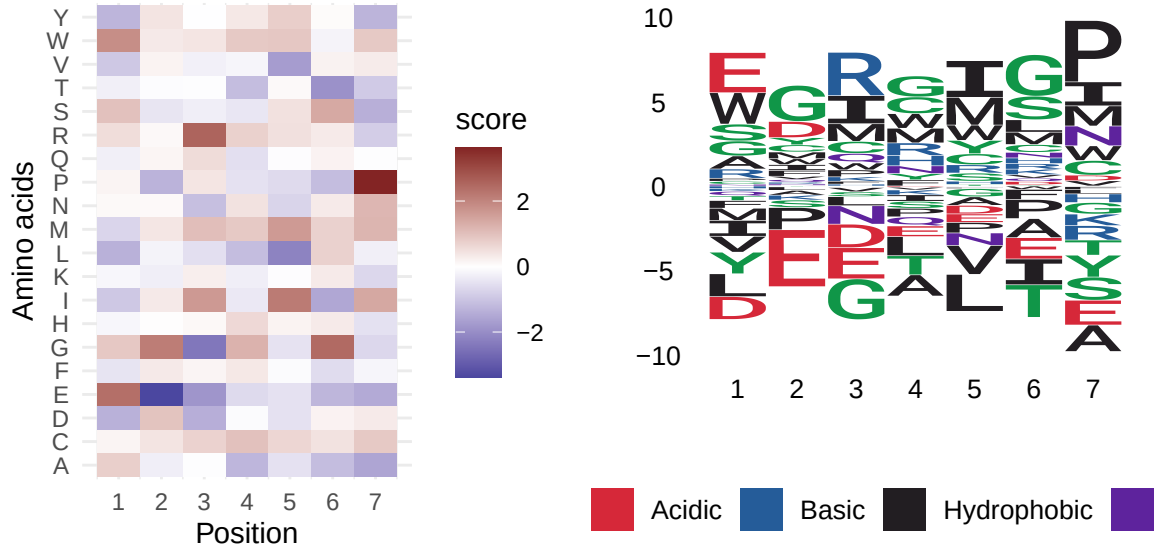
```
diff_pssm_mtx <- compute_diff_pssm(DMF5_pos_peptides, DMF5_neg_peptides)
knitr::kable(diff_pssm_mtx)
```

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| A | 0.7759687 | -0.2869920 | -0.0385592 | -1.2578142 | -0.5066551 | -1.1315838 | -1.5669178 |
| C | 0.1769435 | 0.4254586 | 0.7143191 | 0.9921858 | 0.6549953 | 0.4645008 | 0.8698357 |
| D | -1.3539050 | 0.9583067 | -1.3937402 | -0.0703130 | -0.5066551 | 0.1988590 | 0.3313573 |
| E | 2.3813443 | -3.3742776 | -1.8454672 | -0.6328142 | -0.5066551 | -1.2638462 | -1.4702491 |
| F | -0.4572393 | 0.3462974 | 0.1674369 | 0.3671858 | -0.0701055 | -0.5991833 | -0.1655871 |
| G | 0.8870574 | 2.1420903 | -2.4200596 | 1.2422148 | -0.4866738 | 2.4625850 | -0.6832985 |
| H | -0.1591307 | 0.0296527 | 0.1120164 | 0.6171858 | 0.1903352 | 0.3322384 | -0.5107280 |
| I | -0.9433037 | 0.3462974 | 1.6648304 | -0.3828142 | 2.1725867 | -1.5302326 | 1.4219147 |
| K | -0.1591307 | -0.2869920 | 0.2625921 | -0.2890636 | -0.0419949 | 0.3314938 | -0.6832985 |
| L | -1.3525010 | -0.2078308 | -0.5420454 | -1.1328142 | -2.1891869 | 0.7357272 | -0.2694222 |
| M | -0.7192543 | 0.4002828 | 1.0431807 | 0.8671858 | 1.6442597 | 0.7312595 | 1.1942096 |
| N | 0.1024163 | 0.0872721 | -1.0925889 | 0.4609364 | -0.7389852 | 0.3329830 | 1.1734425 |
| P | 0.1769435 | -1.3160872 | 0.4131678 | -0.5078142 | -0.6228202 | -1.1312115 | 3.6373973 |
| Q | -0.2711554 | 0.1627993 | 0.5637434 | -0.5390636 | -0.0419949 | 0.1988590 | -0.0345507 |
| R | 0.5505152 | 0.1088139 | 2.5682844 | 0.7421858 | 0.4907384 | 0.3314938 | -0.8558690 |
| S | 0.9995501 | -0.4453143 | -0.2842901 | -0.4140636 | 0.4826092 | 1.3977840 | -1.3735803 |
| T | -0.2711554 | -0.0495084 | -0.0385592 | -1.1328142 | 0.0941514 | -1.9296261 | -0.8974030 |
| V | -0.9433037 | 0.1879751 | -0.2565799 | -0.1640636 | -1.7163975 | 0.2018375 | 0.3105902 |
| W | 1.8587183 | 0.3462974 | 0.4131678 | 0.8671858 | 0.9073068 | -0.2005344 | 0.8698357 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Y | -1.2793779 | 0.4254586 | -0.0108490 | 0.3671858 | 0.7911417 | 0.0665966 | -1.2976786 |

Enrichment scores can be visualized by heatmap or sequence logo.

```
plot_raster_diff_pssm(diff_pssm_mtx)
plot_seqlogo_diff_pssm(diff_pssm_mtx)
```




## N-grams

We generate all possible n-grams from input peptides [5], count number of Positive and Negative peptides containing the n-grams, normalize frequency by total number of Positive and Negative peptides respectively, and compute ration-gram = normalized # of Positive peptides containing the n-gram / normalized # of Negative peptides containing the n-gram. `ngram_lengths` parameters allows users to use n i.e. `ngram_lengths = c(2, 3, 4, 5)` means compute statistics for all 2/3/4/5-grams.

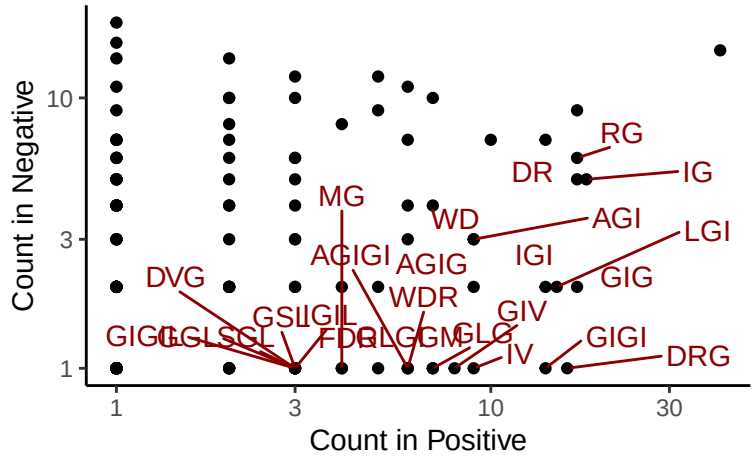
```
ngram_df <- compute_ngrams(DMF5_pos_peptides, DMF5_neg_peptides, ngram_lengths = c(2, 3, 4, 5))
knitr::kable(head(ngram_df, 5))
```

| ngrams | pos_freq | pos_prop | neg_freq | neg_prop | ratio |
|---|---|---|---|---|---|
| G I | 41 | 0.1242424 | 15 | 0.0125000 | 9.939394 |
| I G | 18 | 0.0545455 | 5 | 0.0041667 | 13.090909 |
| D R | 17 | 0.0515152 | 5 | 0.0041667 | 12.363636 |
| L G | 17 | 0.0515152 | 9 | 0.0075000 | 6.868687 |
| R G | 17 | 0.0515152 | 6 | 0.0050000 | 10.303030 |

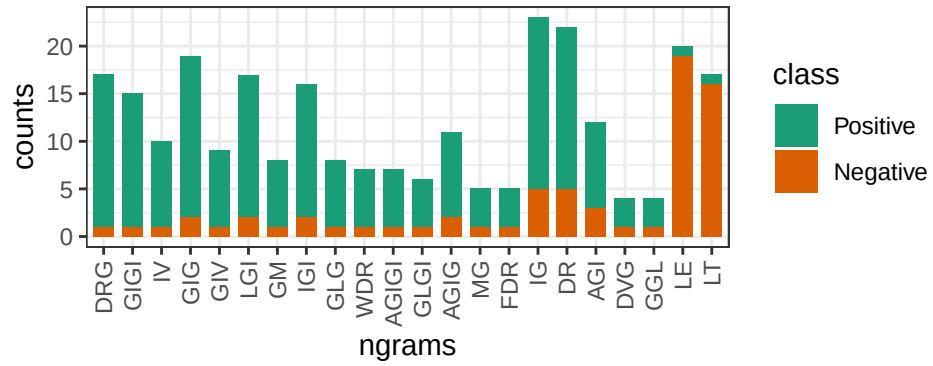
Frequencies of n-grams in Positive and Negative peptide lists can be visualized by scatter plot. Users can denote `ratio_threshold` and `n_threshold` to change thresholds to label n-grams. For n-grams that are present in both Positive and Negative groups, `ratio_threshold` to denote enrichment and depletion ratio to label n-grams, i.e. `ratio_threshold = 4` means only show n-grams that are 4x enriched or depleted. For

n-grams that are only present in either Positive or Negative peptide list, `n_threshold` denotes number of peptides containing the n-gram i.e. n_threshold = 4 means only show n-grams that are present in equal to or more than 4 peptides.

```
plot_point_ngrams(ngram_df, ratio_threshold = 10, n_threshold = 10)
```




```
plot_bar_top_ngrams(ngram_df, top_n = 20)
```



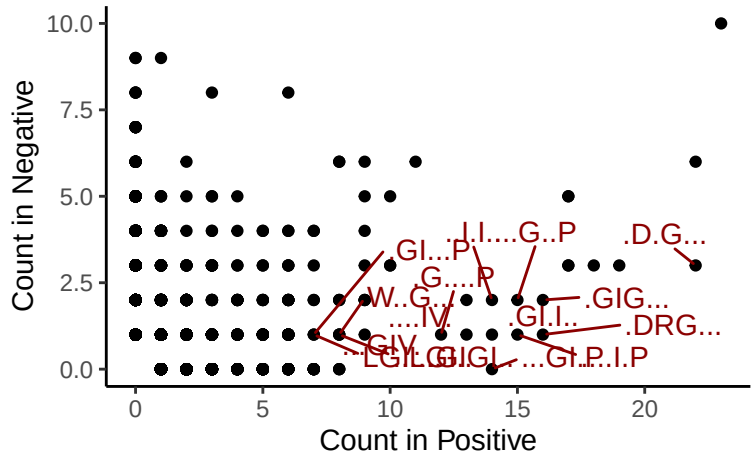
## Position-specific k-mer motifs

A position-specific k-mer is contiguous or non-contiguous sequence of k amino acids, where e.g. .M.W. denotes MW pattern at P2 and P4 of 5 amino acid peptides. Due to position specificity, this analysis is restricted to peptides of same lengths. Similar to n-grams, we compute ratiok-mer = normalized # of Positive peptides containing the positional k-mer / normalized # of Negative peptides containing the positional k-mer.

```
positional_kmer_data <- compute_positional_kmers(DMF5_pos_peptides, DMF5_neg_peptides)
knitr::kable(head(positional_kmer_data, 5))
```

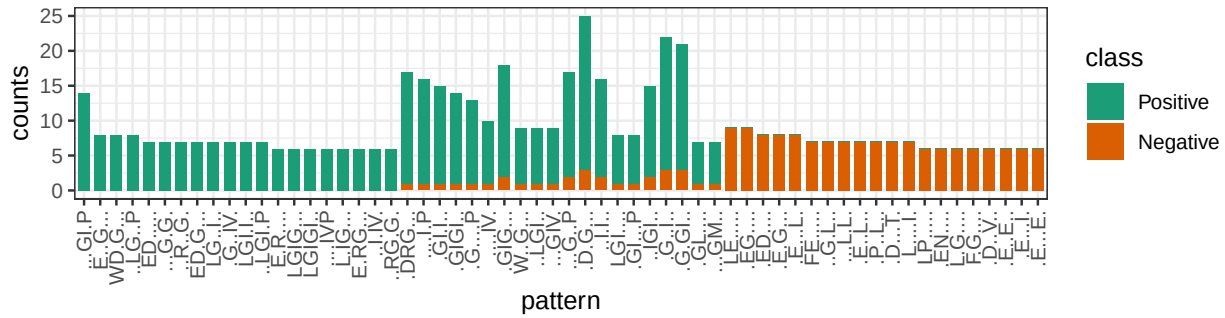
| pattern | count_neg | count_pos | normCount_neg | normCount_pos | ratio |
|---|---|---|---|---|---|
| . . . GI.P | 0 | 14 | 0 | 0.2545455 | Inf |
| E..G. . . | 0 | 8 | 0 | 0.1454545 | Inf |
| WD.G. . . | 0 | 8 | 0 | 0.1454545 | Inf |
| ..LG..P | 0 | 8 | 0 | 0.1454545 | Inf |
| ED. . . .. | 0 | 7 | 0 | 0.1272727 | Inf |

```
plot_point_positional_kmers(positional_kmer_data, ratio_threshold = 25,  n_threshold = 10)
```




Top differential position-specific k-mers can be visualized by barplot. Users can denote top # of k-mers to plot by changing `top_n` parameter.

```
plot_bar_posiitonal_kmers(positional_kmer_data, top_n = 20)
```



## Inter-sequence distance or alignment score

To analyse sequence homology or evolutionary relationship between peptides, users can compute pairwise distance (e.g. Hamming, levenshtein, Jaccard distance) or global/local alignment using substitution matrix of interest (e.g. BLOSUM62) and compare inter-sequence distance between two groups. The options for distance metric are available in `stringdist` package and alignment score in `pairwiseAlignment` function in `Biostring` package.

For illustration purpose, `DMF5_pos_peptides` and only first 60 `DMF5_neg_peptides` were put as input sequence for pairwise alignment score and `DMF5_pos_peptides` for distance.

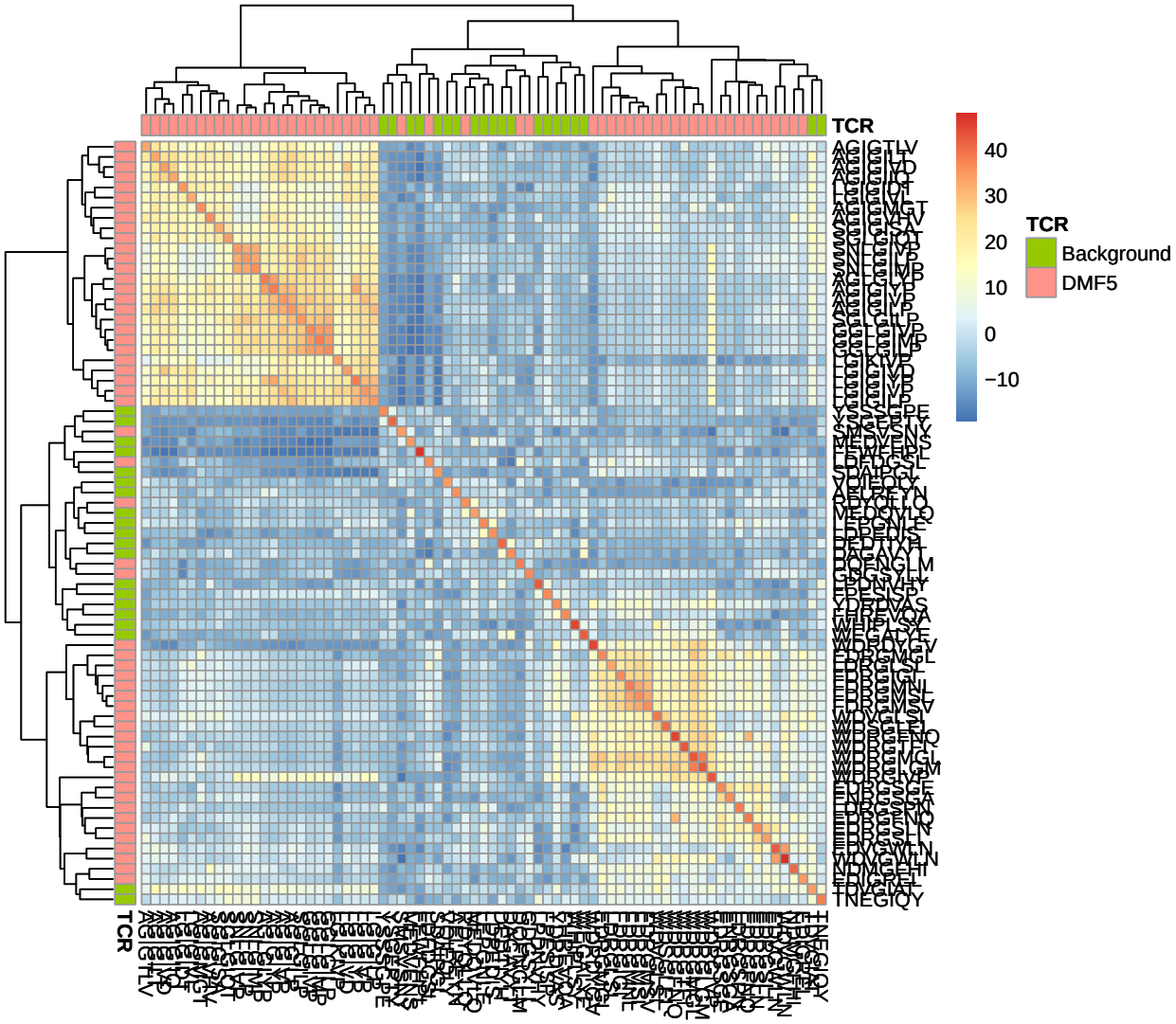
```
alignment_matrix <- compute_pairwise_alignment(peptides = c(DMF5_pos_peptides, DMF5_neg_peptides[1:20]))
distance_mtx <- compute_pairwise_distance(peptides = c(DMF5_pos_peptides))

knitr::kable(alignment_matrix[1:8, 1:8])
```

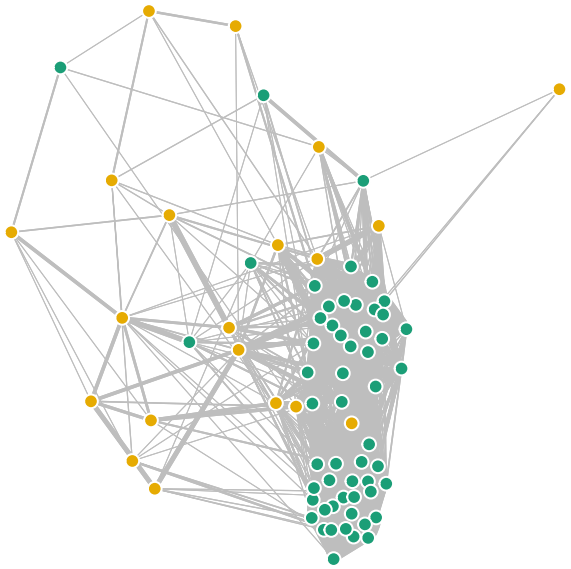
| | LGIGIVP | AGIGIVD | GGLGIMP | SNLGILP | LGIGIYP | AGIGVHV | AGIGTLV | SGLGILP |
|---|---|---|---|---|---|---|---|---|
| LGIGIVP | 35 | 22 | 22 | 18 | 30 | 13 | 13 | 24 |
| AGIGIVD | 22 | 34 | 18 | 13 | 17 | 17 | 17 | 19 |
| GGLGIMP | 22 | 18 | 38 | 23 | 20 | 13 | 13 | 29 |
| SNLGILP | 18 | 13 | 23 | 35 | 16 | 7 | 10 | 29 |
| LGIGIYP | 30 | 17 | 20 | 16 | 38 | 18 | 11 | 22 |
| AGIGVHV | 13 | 17 | 13 | 7 | 18 | 36 | 21 | 13 |
| AGIGTLV | 13 | 17 | 13 | 10 | 11 | 21 | 33 | 16 |
| SGLGILP | 24 | 19 | 29 | 29 | 22 | 13 | 16 | 35 |

Clusters of peptides sharing high sequence homology or evolutionary relationship can be visualized by heatmap or network graph. Users can choose `distance_threshold` or `alignment_threshold` in `plot_network_distance_mtx` or `plot_network_alignment_mtx`, respectively to denote the threshold to plot edges between peptides.

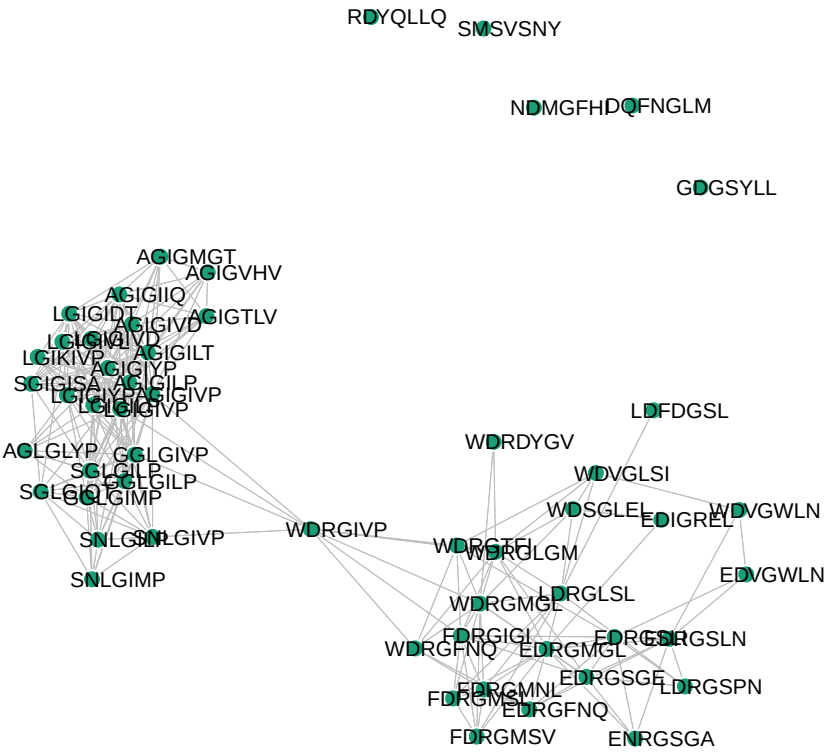
```
library(tibble)
library(dplyr)
col_data <- DMF5_antigen_table %>% column_to_rownames(var = 'ContactPositions' )
plot_heatmap_alignment_mtx(alignment_matrix, col_data)
#plot_heatmap_distance_mtx(distance_mtx, col_data)
```


TCR
40
30
20
10
0
−10
TCR
Background
DMF5
TCR

```
plot_network_alignment_mtx(alignment_matrix, data = DMF5_antigen_table, # need to change data
                           peptide_id_col = 'ContactPositions',  color_col = 'TCR',
                           alignment_threshold = 0,
                           vertex.label.degree = 0,  edge.weight = 0.2,
                           vertex.size= 5,  vertex.label.cex = 0.7)
#> NULL
```



```
plot_network_distance_mtx(distance_mtx, data = DMF5_antigen_table,
                           peptide_id_col = 'ContactPositions',  color_col = 'TCR',
                           distance_threshold = 3,
                           vertex.label.degree = 0,  edge.weight = 0.2,
                           vertex.size= 5,  vertex.label.cex = 0.7, label_vertex=TRUE)
#> NULL
```




## References

[1] Gee, M. H. et al. Antigen Identification for Orphan T Cell Receptors Expressed on Tumor-Infiltrating Lymphocytes. Cell 172, 549-563.e16 (2018).

[2] Joglekar, A. V. et al. T cell antigen discovery via signaling and antigen-presenting bifunctional receptors. Nature Methods 16, 191–198 (2019).

[3] Calis, J. J. A. et al. Properties of MHC Class I Presented Peptides That Enhance Immunogenicity. PLOS Computational Biology 9, e1003266 (2013).

[4] Pagès, H., Aboyoun, P., Gentleman, R. & DebRoy, S. Biostrings: Efficient manipulation of biological strings. (Bioconductor version: Release (3.14), 2022). doi:10.18129/B9.bioc.Biostrings.

[5] Schmidt, D. & Heckendorf, C. ngram: Fast n-Gram 'Tokenization'. (2021).

## Citation

To cite this package, please use:

```
citation('diffSeqPatterns')
#>
#> To cite package 'diffSeqPatterns' in publications use:
#>
#>   Chloe H. Lee (NA). diffSeqPatterns: Differential peptide sequence patterns between two groups. R
#>   package version 0.1.0.
#>
#> A BibTeX entry for LaTeX users is
#>
#>   @Manual{,
#>     title = {diffSeqPatterns: Differential peptide sequence patterns between two groups},
#>     author = {Chloe H. Lee},
#>     note = {R package version 0.1.0},
#>   }
#>
#> ATTENTION: This citation information has been auto-generated from the package DESCRIPTION file and
#> may need manual editing, see 'help("citation")'.
```

## SessionInfo

```
sessionInfo()
#> R version 4.0.5 (2021-03-31)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 19042)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=English_United States.1252  LC_CTYPE=English_United States.1252
#> [3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
#> [5] LC_TIME=English_United States.1252
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] dplyr_1.0.7           tibble_3.1.2          diffSeqPatterns_0.1.0
#>
#> loaded via a namespace (and not attached):
#>  [1] ggrepel_0.9.1        Rcpp_1.0.6           stringdist_0.9.7     lubridate_1.7.10     lattice_0.2
#>  [6] class_7.3-19         Biostrings_2.56.0    ggseqlogo_0.1        digest_0.6.27        assertthat_
#> [11] ipred_0.9-11         foreach_1.5.1        utf8_1.2.1           R6_2.5.0             plyr_1.8.6
#> [16] stats4_4.0.5         evaluate_0.14        highr_0.9            ggplot2_3.3.5        pillar_1.6.
#> [21] zlibbioc_1.34.0      PepTools_0.1.0       rlang_0.4.11         caret_6.0-88         rstudioapi_
#> [26] data.table_1.14.0    S4Vectors_0.26.1     rpart_4.1-15         Matrix_1.3-4         rmarkdown_2
#> [31] labeling_0.4.2       splines_4.0.5        gower_0.2.2          stringr_1.4.0        igraph_1.2.
#> [36] pheatmap_1.0.12      munsell_0.5.0        xfun_0.29            compiler_4.0.5       pkgconfig_2
#> [41] BiocGenerics_0.34.0  htmltools_0.5.1.1    nnet_7.3-16          tidyselect_1.1.1     prodlim_201
#> [46] IRanges_2.22.2       codetools_0.2-18     reshape_0.8.8        fansi_0.5.0          crayon_1.4.
#> [51] withr_2.4.2          MASS_7.3-54          recipes_0.1.16       ModelMetrics_1.2.2.2 grid_4.0.5
#> [56] nlme_3.1-152         gtable_0.3.0         lifecycle_1.0.0      DBI_1.1.1            magrittr_2.
#> [61] pROC_1.17.0.1        scales_1.1.1         stringi_1.6.2        farver_2.1.0         XVector_0.2
#> [66] reshape2_1.4.4       remotes_2.4.0        timeDate_3043.102    ellipsis_0.3.2       generics_0.
#> [71] vctrs_0.3.8          lava_1.6.9           RColorBrewer_1.1-2   iterators_1.0.13     tools_4.0.5
#> [76] glue_1.4.2           purrr_0.3.4          yaml_2.2.1           parallel_4.0.5       survival_3.
#> [81] colorspace_2.0-2     ngram_3.1.0          knitr_1.33
```