Introduction to *Palimpsest*

Jayendra Shinde [*1] ***and Eric Letouzé*** [†1]

[1]INSERM, UMR-1162, Génomique Fonctionnelle des Tumeurs Solides, Equipe Labellisée Ligue Contre le Cancer, Institut Universitaire d'Hématologie, Paris, France

*jayendra.shinde91@gmail.com †eric.letouze@inserm.fr

16 February 2018

Abstract

Cancer genomes are altered by various mutational processes and, like palimpsests, bear the signatures of these successive processes. The Palimpsest R package provides a complete workflow for the characterization and visualization of mutational signatures and their evolution along tumor development. The package covers a wide range of functions for extracting both base substitution and structural variant signatures, inferring the clonality of each alteration and analyzing the evolution of mutational processes between early clonal and late subclonal events. Palimpsest also estimates the probability of each mutation being due to each process to predict the mechanisms at the origin of driver events. Palimpsest is an easy-to-use toolset for reconstructing the natural history of a tumor using whole exome or whole genome sequencing data

Package

Report issues on www.github.com/FunGeST/Palimpsest

Contents

1 Introduction

A cancer genome is made up of a heterogeneous landscape of somatic mutations, often involving large-scale chromosomal alterations. The patterns of mutations within cancer cells, dynamics of clonal diversification and selection are crucial in understanding the natural history of the cancer. Recent advances in sequencing technologies have led to an exponential growth in the genome sequencing studies. Such tremendous rise in sequencing projects has led researchers to study increasingly large, cohort level genomic data. Summarization, interpretation and visualization of this ever-growing data still remain to be a major challenge. Generation of publication-quality figures for efficient communication is often a time consuming step generally involving multiple manual curations. Here, we describe an R package, Palimpsest a comprehensive easy-to-use tool for flexible visualization of mutational signatures, copy number alterations and reconstructing clonal architecture within a given tumor sample.

2 Installation Instructions

The package can be installed from the GitHub repository using `devtools`:

```
----------------------------------------------------------------------
>library(devtools)
>devtools::install_github("FunGeST/Palimpsest")
----------------------------------------------------------------------
```

3 Dependencies

Recommended to install R package "bedr" in order to perform structural variant signature analysis. The bedr API gives access to "BEDTools" as well as offers additional utilities for genomic regions processing. To gain the functionality of bedr package you will need to have the BEDTools program installed and in your default PATH.

4 Input Data

The package performs exhaustive analysis as well as visualization of the whole genome sequencing data for comprehension of the activities and characteristics of the mutational processes active within the cancer genomes. The two important input data types as provided in example datasets, necessary to perform the analyses are:
1] mut_data: Mutational catalogue listing SSMs (simple somatic mutations, i.e., single somatic nucleotide variants).
2] cna_data: File listing allele specific copy number information.
3] annot_data: Annotation data consisting of sample gender information and tumor purity.

Optional: The mutational signatures analysis can be extended to structural variants.
4] sv_data: File listing structural variants information

For proper format, please refer the example files provided with the package.
Check out the README file to find more information about the formats of the input files: https://github.com/FunGeST/Palimpsest

4.1 Defining working directories

Set directories in order to set directories containing datasests and to export results generated by the use of Palimpsest.

```
-----------------------------------------------------------------
# directory containing data files
datadir <- "Data/LiC1162"

# directory to export results
resdir <- "Results";if(!file.exists(resdir)) dir.create(resdir)
-----------------------------------------------------------------
```

4.2 Loading genomic data and reference genome

We provide example input datasets along with this package. The accompanying example data is from the paper Mutational signatures reveal the dynamic interplay of risk factors and cellular processes during liver tumorigenesis, with authors Letouzé, E#, Shinde, J.#, Renault, V., Couchy, G., Blanc, J.-F., Tubacher, E., Bayard, Q., Bacq, D., Meyer, V., Semhoun, J., Bioulac-Sage, P., Prévôt, S., Azoulay, D., Paradis, V., Imbeaud, S., Deleuze, J.-F. and Zucman-Rossi, J.

```
-----------------------------------------------------------------
#Load the Palimpsest R library
>library(Palimpsest)

#Loading the example data.
>load(file.path(datadir,"mut_data.RData"))
>load(file.path(datadir,"cna_data.Rdata"))
>load(file.path(datadir,"annot_data.Rdata"))
>load(file.path(datadir,"sv_data.Rdata"))
-----------------------------------------------------------------
```

The package works with the choice of reference genomes available via BSgenome. Download and load your reference genome of interest:

```
-------------------------------------------------------------------
  >library(BSgenome.Hsapiens.UCSC.hg19)
  >ref_genome <- BSgenome.Hsapiens.UCSC.hg19
-------------------------------------------------------------------
```

4.3 Preprocessing and annotating input data

We preprocess the input example data with necessary fields using public dataset and hg19 reference genome. For SNVs, for example we retrieve the neighboring mutation contexts for every base change in the mutations based on its position on the reference genome. Mutational signatures are visualized based on the proportion of presence of mutational categories present in the genome. The SNV signatures are representation of the trinucleotide contexts frequencies in the genome. With the function palimpsestInput(), we can create a vector corresponding to the counts of these mutational categories present in each sample within the cohort.

```
-------------------------------------------------------------------
>vcf <- preprocessInput_snv()

>propMutsByCat <- palimpsestInput()
-------------------------------------------------------------------
```

5 Mutational Signatures

With the input of matrix M made up of the features comprising of mutational counts for each mutation type, contains K mutation types across G samples. By modeling mutational mechanisms as a bind source separation problem, we try to take a glance into the combinations of various mutational processes active within the tumor cohort.

5.1 De-Novo mutational signature analysis using NMF

The De-Novo mutational signature extraction is based on the use of Non-Negative Matrix Factorization provided in the NMF package (Gaujoux & Seoighe, 2010). Optimal factorization rank is a critical parameter for the Non- Negative Matrix Factorization (NMF) algorithm, which is the number of stable clusters observed within the tumor samples. As described in their paper, the method of choosing the right value of rank for NMF is the rank at which the cophenetic correlation co-efficient starts reducing. Alco, another approach is to choose the rank for which the plot of residual sum of squares (RSS) is the maximum between input and it's estimate. Using the flexible function provided, we can automatically decipher the optimal number of mutational signatures observed in the series of tumors based on cophenetic coefficient. Also, we can manually input the number of signatures you'd like to extract from the mutation data. In case of larger data, it is advisable to allot higher number of iterations (nrun) parameter in order to achieve stable number of mutational signatures and avoid local minima.

```
-----------------------------------------------------------------------
>denovo_signatures <-deconvolution_nmf(input_data = propMutsByCat,
                                       type = "SNV",
                                       range_of_sigs =2:10,
                                       nrun =20,
                                       method = "brunet",
                                       resdir = results)
-----------------------------------------------------------------------
```

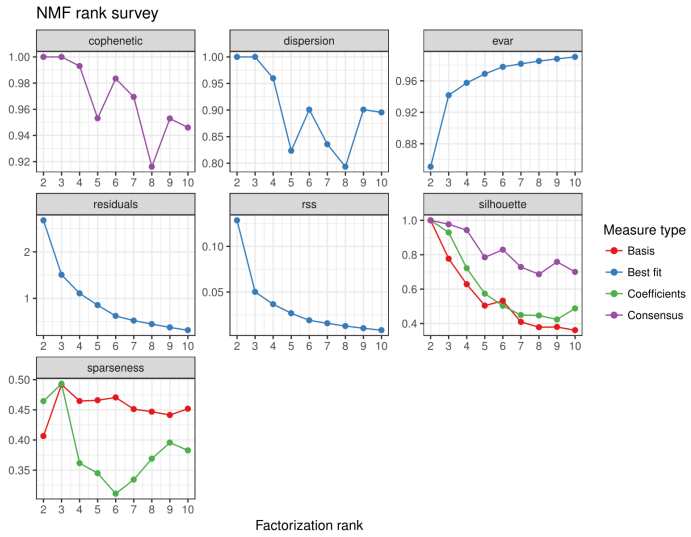

Figure 1: NMF Rank Estimation

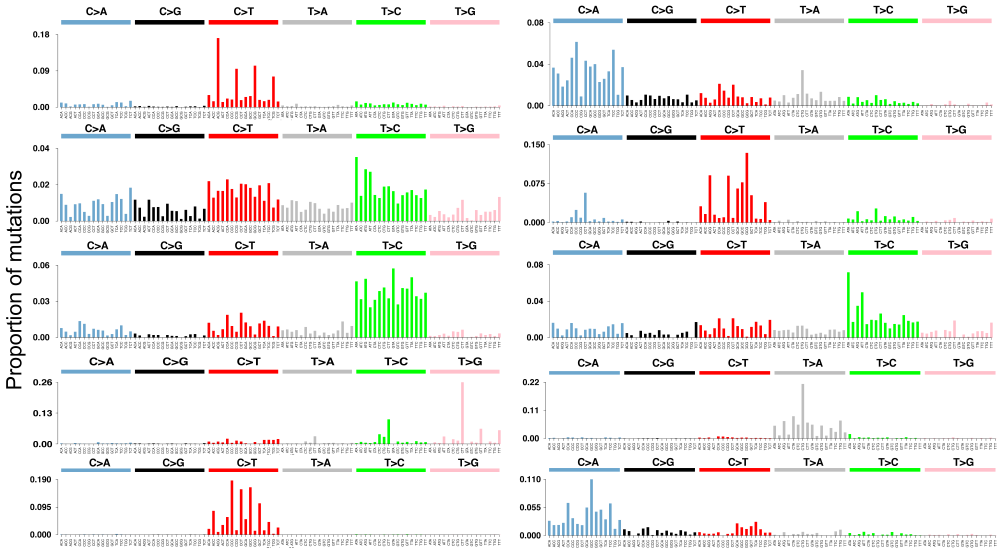

Figure 2: De Novo extraction of mutaional signatures

5.2 Cosine similarity

To find the similarity between the newly extracted signatures and the already described signatures, cosine similarity can be used. This ranges between zero and one, where a similarity of one represents identical signatures and a similarity of zero completely different mutational signatures

```
-----------------------------------------------------------------------
# Compare with existing signatures from COSMIC database

>cosine_similarities <- deconvolution_compare(denovo_signatures,COSMIC_Signatures)
-----------------------------------------------------------------------
```

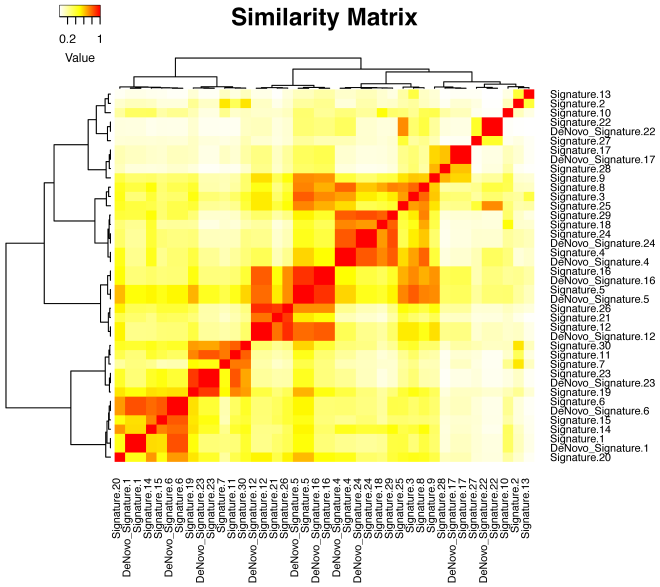

Figure 3: Cosine similarity matrix- Comparing the de novo extracted signatures with previously described mutational signatures (COSMIC)

5.3 Optimal exposure of mutational signatures

The approach is based on analysis in order to determine the contributions of mutational signatures de-novo extracted or previously described (COSMIC). We decompose the mutation matrix M as the product of a matrix P representing the signature of each mutagenic process and a matrix E representing the exposure of each tumor to each process. The P matrix comprised a pre-defined set of signatures previously identified (Alexandrov et al.). We implemented the fast combinatorial strategy approach from the NMF package in order to solve the following nonnegative least squared problem.

min ||P – M*E||_F, s.t. E >=0

where P and M are two matrices of dimension n x p and n x r respectively, and |.|_F is the Frobenius norm.The package provides with the deconvolution_exposure() function for calculating the exposures of input signatures obtained from de-novo extraction or already known signatures as well as plotting these exposures across the cohort.

```
-------------------------------------------------------------------
# Calculating contributions (exposures) of de-novo signatures

>signatures_exp <- deconvolution_fit(vcf=vcf,
                                     type = "SNV",
                                     input_data = propMutsByCat,
                                     threshold = 5,
                                     input_signatures = denovo_signatures,
                                     sig_cols = mycol,
                                           plot = T,
                                           resdir = resdir.)


# Calculating contributions (exposures) of COSMIC liver signatures

>signatures_exp <- deconvolution_fit(vcf=vcf,
                                    type = "SNV",
                                    input_data = propMutsByCat,
                                    threshold = 5,
                                    input_signatures = liver_signatures,
                                    sig_cols = mycol,
                                    plot = T,
                                    resdir = resdir.)
  -------------------------------------------------------------------
```

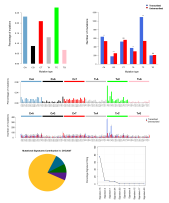

Figure 4: : Fitting extracted mutaitonal signatures in tumor profiles

Calculating the exposure of mutational signatures in each sample lets us visualize their contributions across the series.

```
-----------------------------------------------------------------
# Plotting the exposures of signatures across the series:

>deconvolution_exposure()
-----------------------------------------------------------------
```

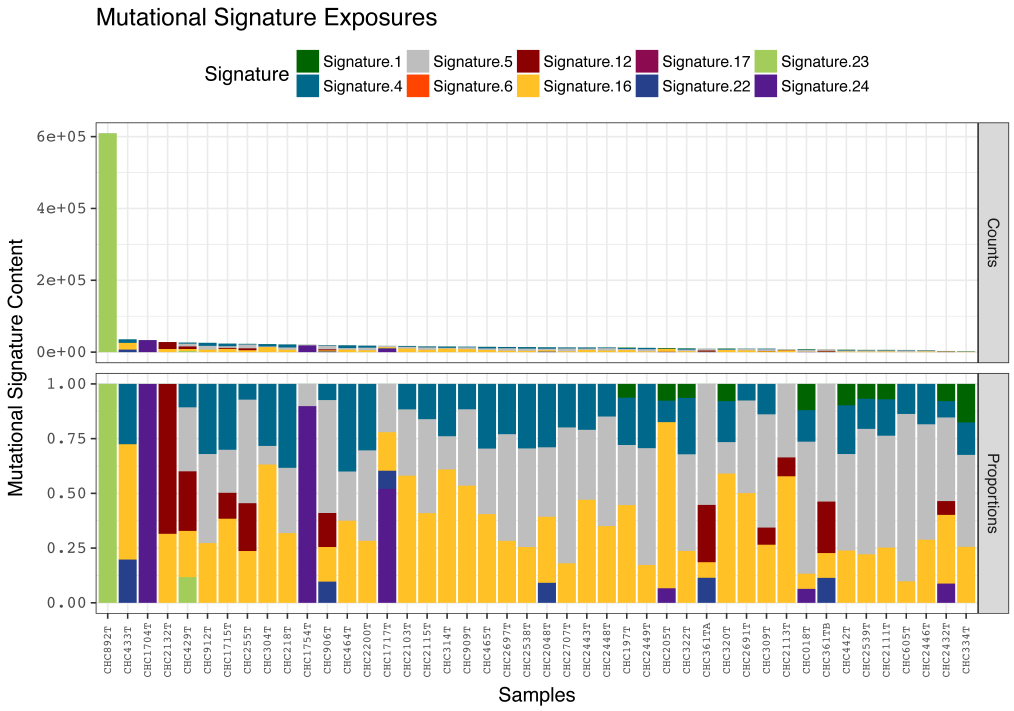

Figure 5: : Contribution of mutational signatures across the series of samples

5.4 Mutation-Signature Classifier

Using a Bayes' classifier, we developed a computational approach to assign the probable source of each mutational event. The package provides a unique function to estimate the probability of each somatic mutation being due to each mutational signature considering the mutation category and the number of mutations attributed to each signature in the corresponding tumor sample. The probability $P(m, s)$ of a mutation m of category c in tumor t being due to signature s out of n signatures can then be estimated as:

$$P(m,s) = \frac{p_s^c \times e_t^s}{\sum_{s=1}^{n} p_s^c \times e_t^s}$$

```
-----------------------------------------------------------------
>vcf <- palimpsestOrigin()
-----------------------------------------------------------------
```

Using the assigned probability values, we can perform estimation of cumulative contribution of signatures to each driver gene. Contribution of mutational signatures in driver gene mutations can lead to insights into identifying genes preferentially mutated by specific mutational processes. Distribution of mutational signatures associated with driver gene mutations from Letouzé, Shinde et.al can be performed using this method.

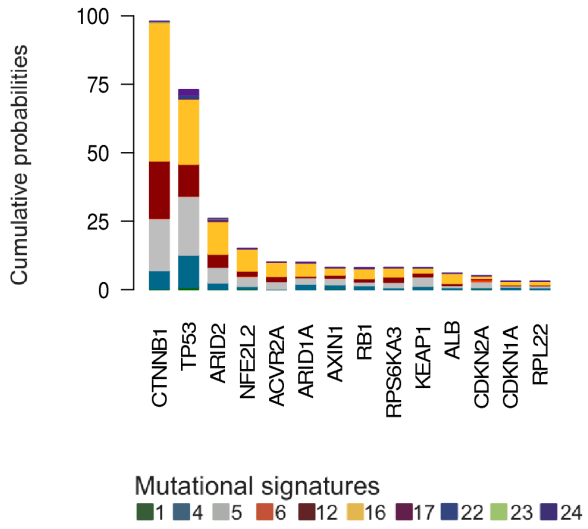

Figure 6: : Distribution of mutational signatures associated with driver gene mutations across the series of 299 HCC whole genomes represented as cummulative probabilities for each driver gene.

5.5 Rainfall plots

In order to explore regional variants in mutation rates and clustering of mutations across the cancer genome, we introduce an additional utility function rainfallz_plot() in order to visualize the "rainfall plots". The y-axis represents the ilog base 10 scale of ntermutational distance while the x-axis deontes the ordered mutations along the genomic coordinates.

```
-----------------------------------------------------------------
>rainfallz_plot()
-----------------------------------------------------------------
```

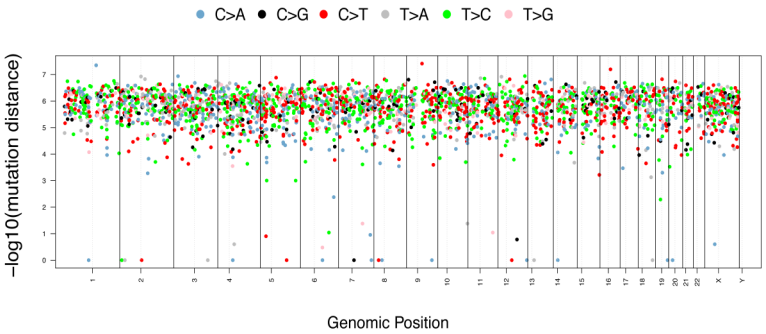

Figure 6: Rainfall Plot

6 Clonality Analysis

The package provides a series of functions in order to combine the copy number information and somatic mutations data in order to reconstruct the natural history of a tumor.

6.1 Copy number alterations and Cancer cell fraction (CCF)

The function cnaCCF_annot() provies automated steps for calculating the cancer cell fraction using the copy number information supplied as input. The function estimates the proportion of tumor cells harboring the mutation (cancer cell fraction, CCF) using the method described in Letouzé, Shinde et.al

$CCF = VAF \times \frac{\rho N_t + (1-\rho) N_n}{\rho n_{chr}}$

where ρ is the tumor purity,N_t and N_n the copy-number at the locus in tumor and normal cells, and n_{chr} the number of chromosomal copies harboring the mutation in tumor cells.The function actively associates every mutation in the input vcf with the chromosome arm copy number and assigns its respective CCF information.

```
----------------------------------------------------------------------
>vcf <- cnaCCF_annot()
----------------------------------------------------------------------
```

6.2 Clonality plots

The plotting function provides with a series of functionalities neceaasry for visualization of the cancer genome clonality information as per annotated in the preceeding section.

```
----------------------------------------------------------------------
>cnaCCF_plots()
----------------------------------------------------------------------
```

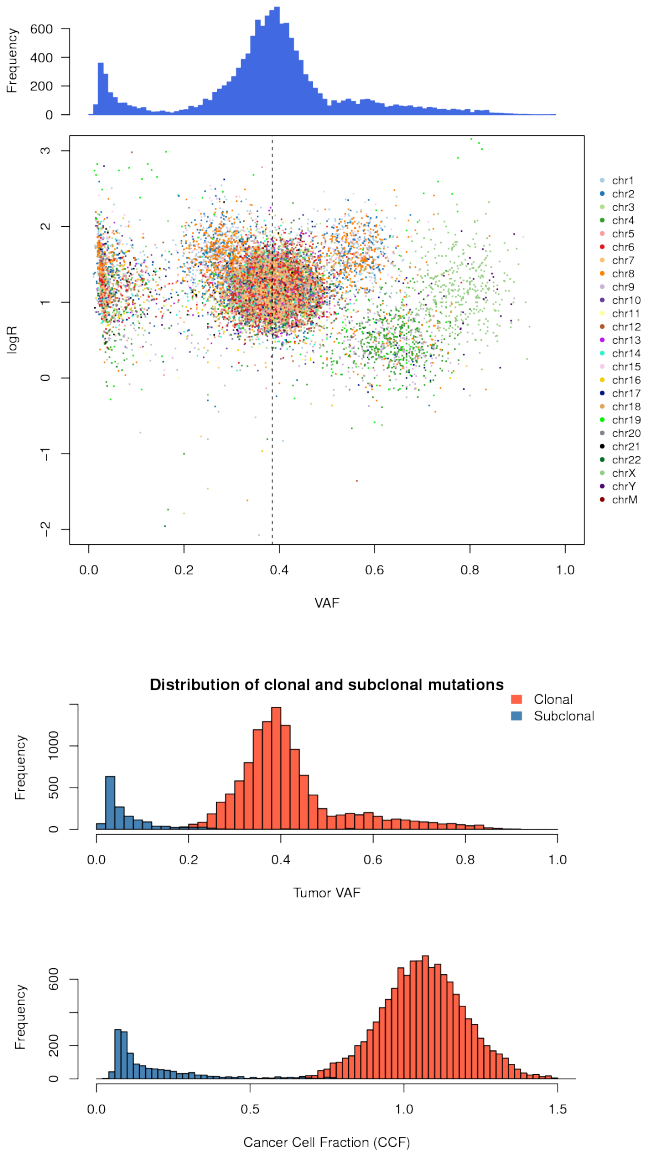

Figure 7: Cancer genome variant allele fraction and Cancer cell fraction (CCF)

6.3 Temporal dissection of mutational signatures

In order to further our understanding of temporal clonal evolutionary processes, we use the defined clonal and subclonal compartments within a cancer genome in order to dissect different mutational processes operative in these compartments. We represent the 96-nucleotide type profiles representative of clonal as well as subclonal mutations in a genome. Using these we aare able to perform mutational signature analysis and calculate the contribution of different mutational signatures moulding the cancer's evolutionary process.

```
-----------------------------------------------------------------------
>palimpsest_DissectSigs()
-----------------------------------------------------------------------
```

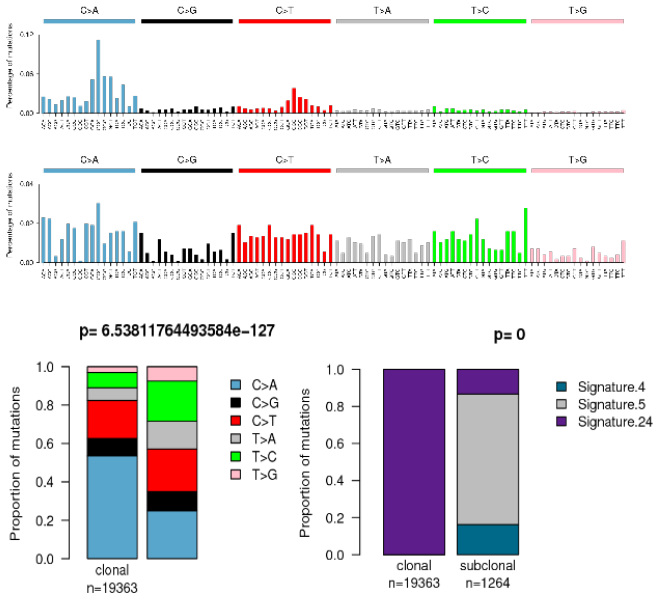

Figure 8: Temporal dissection of mutational signatures

Proportion of mutations attributed to each mutational signature in the clonal and subclonal mutations of each tumor across the series can be conveniently visualized by use of the following function:

```
-----------------------------------------------------------------------
>palimpsest_clonalitySigsCompare(clonsig = signatures_exp_clonal$sig_nums,
                                 subsig = signatures_exp_subclonal$sig_nums,
                                 msigcol = mycol,
                                 resdir = results)
-----------------------------------------------------------------------
```

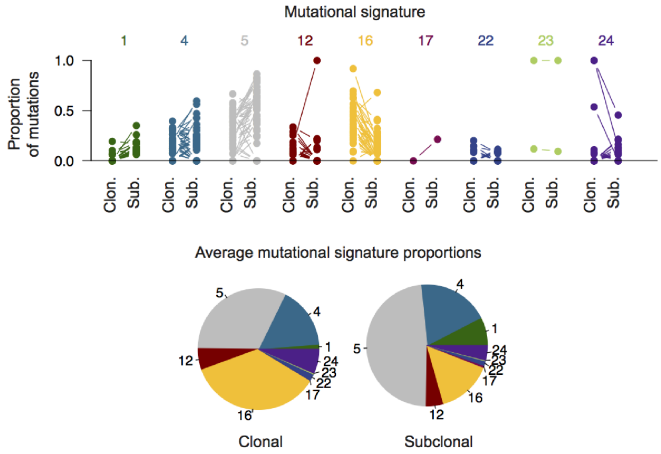

Figure 8: Proportion of mutations associated with mutational signatures in clonal and subclonal compartments of cancer genomes across the series of tumors.

6.4 Timing chromosomal gains

We use the number of duplicated and non-duplicated mutations to estimate the timing of each chromosome duplication. Consider the simple case of a chromosome with absolute copy-number Nt=3. The molecular time at which the extra copy of the chromosome was gained can be estimated as:

$T = N_{dup}/[N_{dup} + [(N_{ndup} - N_{dup})/3]]\ddot{O}100/$

where $N_{dup}/$ and $N_{ndup}/$ are the number of duplicated and non-duplicated mutations, respectively.

```
-----------------------------------------------------------------------
>time_Data <- chrTime_annot()
>chrTime_plot()
-----------------------------------------------------------------------
```

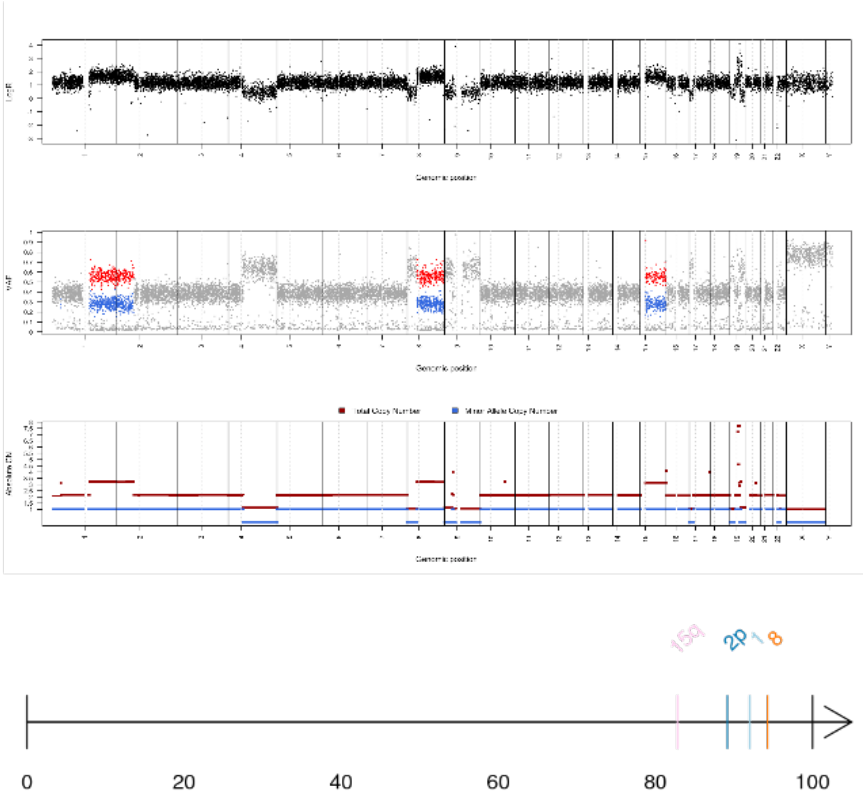

Figure 9: Chromosomal duplication timeline

7 Structural Variants (SV) Signatures:

Palimpsest provides extraction of mutational processes underlying structural variants (SVs) including CIRCOS plot representation of each tumor profile and extraction of structural variant signatures using the same statistical approach used for substitution signature analysis described above. SV events are defined into a total of 38 categories of structural variants considering the type (deletion, tandem duplication, inversion, interchromosomal translocation) and size (<1kb, 1-10kb, 10-100kb, 100kb-1Mb, 1-10Mb, >10Mb) of rearrangements.

```
-----------------------------------------------------------------------
>library(bedr) # dependency for categorizing SV events
>vcf <- preprocessInput_sv(input_data =  sv_file,
                           ensgene = ensgene,
                           resdir = resdir.)
>input_data <- palimpsestInput()
>input_signatures <- deconvolution_nmf()
-----------------------------------------------------------------------
```

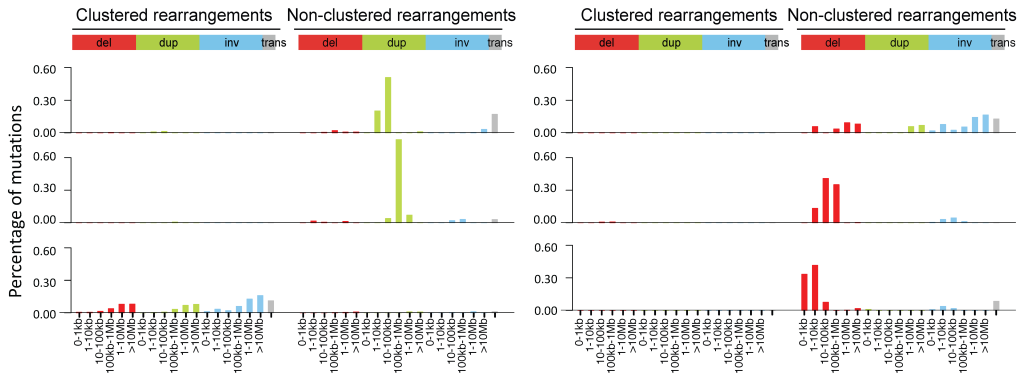

Figure 10: Structural variants signatures

8 Natural history of tumors

Palimpsest provides with a set of functions for visualization of the evolutionary history of individual tumors giving important insights on the catalogue of somatic mutations they have accumulated over time.

Using clonal and subclonal signature contributions, Palimpsest helps in visualizing the temporal dissection of mutational/ structural signatures within a cancer genome ultimately reconstructing it's oncogenic timeline.

```
-----------------------------------------------------------------
>library(bedr) # dependencies for categorizing SV events
>palimpsest_plotTumorHistories(vcf = vcf,
                               sv.vcf = NULL,
                               cna_data,
                               point.mut.time,
                               clonsig=signatures_exp_clonal$sig_props,
                               subsig=signatures_exp_subclonal$sig_props,
                               msigcol=mycol,
                               msigcol.sv=mycol.sv,
                               resdir=resdir.)
-----------------------------------------------------------------
```

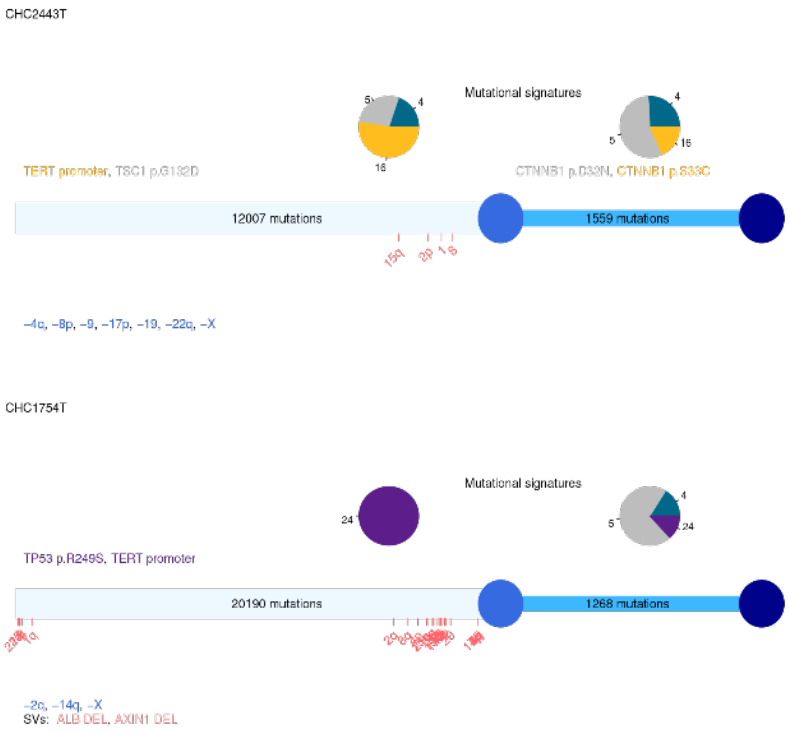

Figure 10: Oncogenic timelines representing the natural history of tumors

9 References

1. Alexandrov, L. B. et al. Signatures of mutational processes in human cancer. Nature 500, 415–21 (2013).
2. Alexandrov, L. B., Nik-zainal, S., Wedge, D. C., Campbell, P. J. & Stratton, M. R. Deciphering Signatures of Mutational Processes Operative in Human Cancer. CellReports 3, 246–259 (2012).
3. Gaujoux, R. & Seoighe, C. A flexible R package for nonnegative matrix factorization. BMC Bioinformatics 11, 367 (2010).
4. Nik-Zainal, S. et al. The life history of 21 breast cancers. Cell 149, 994–1007 (2012).
5. Nik-Zainal, S. et al. Landscape of somatic mutations in 560 breast cancer whole-genome sequences. Nature 534, 47–54 (2016).
6. Letouzé, E., Shinde, J. et al. Mutational signatures reveal the dynamic interplay of risk factors and cellular processes during liver tumorigenesis. Nature Commun. 8(1):1315. (2017)

Session info

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.3 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] BiocStyle_2.6.1
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.4.3  backports_1.1.2 bookdown_0.6    magrittr_1.5
##  [5] rprojroot_1.3-2 tools_3.4.3     htmltools_0.3.6 yaml_2.1.16
##  [9] Rcpp_0.12.15    stringi_1.1.6   rmarkdown_1.8   knitr_1.19
## [13] xfun_0.1        stringr_1.2.0   digest_0.6.15   evaluate_0.10.1
```