Introduction to *Palimpsest*

***Jayendra Shinde*[*1], *Quentin Bayard*[1], *Sandrine Imbeaud*[1], *Théo Hirsch*[1], *Feng Liu*[2], *Victor Renault*[2], *Jessica Zucman-Rossi*[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.
[2]Laboratory for Bioinformatics, Fondation Jean Dausset – CEPH, Paris, France.

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

21 March 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

This document presents a typical analysis of whole genome sequencing data using Palimpsest. The corresponding script (Palimpsest.R) and data (LiC1162) are provided with the package for testing. This script allows to (1) extract de novo or known mutational signatures, (2) estimate the probability of each mutation being due to each process, (3) infer the clonality of each mutation and compare early and late mutational signatures, (4) estimate the timing of chromosome duplications using somatic mutations, (5) analyze structural variant signatures and (6) represent all these results in a schematic tumor history plot.

2 Installation Instructions

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

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

3 Dependencies

The "bedr" package is required to perform structural variant signature analysis. The bedr API gives access to "BEDTools" and 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

Three input files are necessary to perform the core Palimpsest analyses:
1] mut_data: Somatic mutation catalogue of the tumor series.
2] cna_data: Segmented copy-number data.
3] annot_data: Minimal sample annotation data (including gender and tumor purity).

Optional: The mutational signature analysis can be extended to structural variants.
4] sv_data: Somatic structural variants identified in the tumor series.

For extended format descriptions please refer to the example files provided with the package and check out the README file: https://github.com/FunGeST/Palimpsest

4.1 Loading genomic data and reference genome

We first define the directory containing input data and the directory where result files should be exported.

```
----------------------------------------------------------------------
# directory containing example dataset
datadir <- "Palimpsest/RUNNING_PALIMPSEST_EXAMPLE/LiC1162"

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

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, by Letouzé#, Shinde# et al.

```
----------------------------------------------------------------------
#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. Make sure you have the BSgenome library installed and load the appropriate reference genome. A gene table (ensgene) and cytoband table (cyto) are also required. Hg19 and hg38 versions of these tables are provided with the package in data folder.

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

  >load("Palimpsest/data/ensgene_hg19.RData")
  >load("Palimpsest/data/cytoband_hg19.RData")
-------------------------------------------------------------------
```

4.2 Preprocessing and annotating input data

Mutation data preprocessing involves retrieving the substitution type and trinucleotide context from the reference genome, as well as the transcribed strand information for transcriptional strand bias analyses. A gene table is required for transcriptional strand annotation. The table of Ensembl genes (ensgene) is provided with the package in hg19 and hg38 formats. Mutation data are then converted as a matrix propMutsByCat giving the proportion of each mutation category in each sample that serves as input for NMF.

```
-------------------------------------------------------------------
>vcf <- preprocessInput_snv(input_data = mut_data,
                            ensgene = ensgene,
                            reference_genome = ref_genome)

># Processing input for mutational signature extraction
>propMutsByCat <- palimpsestInput(vcf = vcf,
                                  type = "SNV",
                                  sample.col = "Sample",
                                  mutcat.col = "mutcat3",
                                  proportion = TRUE)
-------------------------------------------------------------------
```

5 Mutational Signatures

Palimpsest offers two ways to analyze mutational signatures. A first option is to perform de novo mutational signature analysis. In this case, non-negative matrix factorization (NMF) is used to estimate the number of processes operative in the data, the signature of each process and its activity in each tumor. This approach allows to identify new mutational signatures. Another option is to extract known signatures (e.g. the consensus signatures from COSMIC database). In this case NMF is only used to estimate the activity of each process in each tumor.

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). The factorization rank (i.e. the optimal number of signatures) can be manually defined by the user or estimated automatically considering the cophenetic correlation coefficients and residual sum of squares (RSS), as described in the original article. For large datasets, it is advisable to allot higher number of iterations (nrun) parameter to avoid local minima and obtain a stable number of mutational signatures.

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

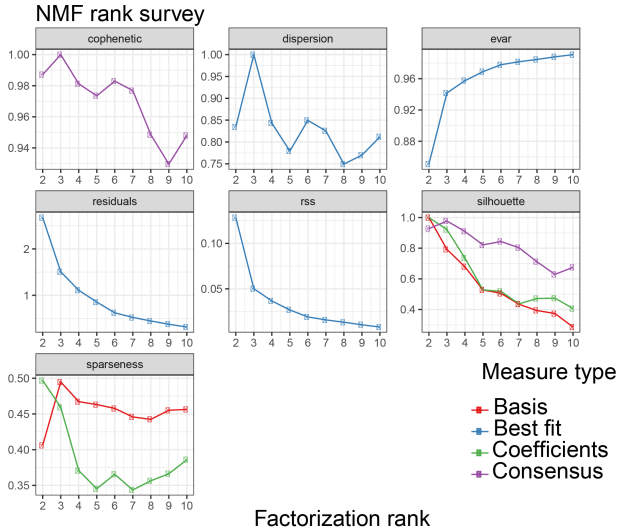

Figure 1: NMF rank estimation. These outputs of the NMF package allow the user to determine the best number of signatures. Otherwise Palimpsest will do it automatically taking the last rank before the cophenetic distance starts reducing.

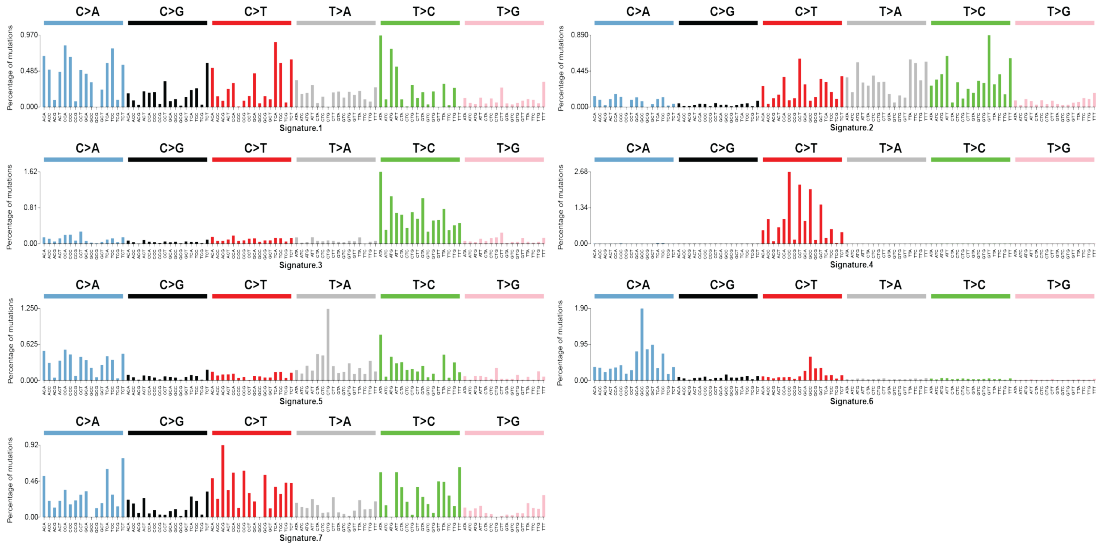

Figure 2: Mutational signatures extracted by de novo NMF analysis. Here, 7 signatures were identified. Each mutational signature is represented as a barplot giving the frequency of mutations in each category, taking into account substitution types (top) and trinucleotide contexts (bottom).

5.2 Cosine similarity

Once de novo mutational signatures have been extracted, it is useful to determine whether they represent new mutational processes or correspond to previously described signatures. The cosine_similarities function estimates the cosine similarity score (0 = completely different, 1 = identical) between the extracted signatures and a set of known signatures (e.g. COSMIC signatures provided with the package).

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

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

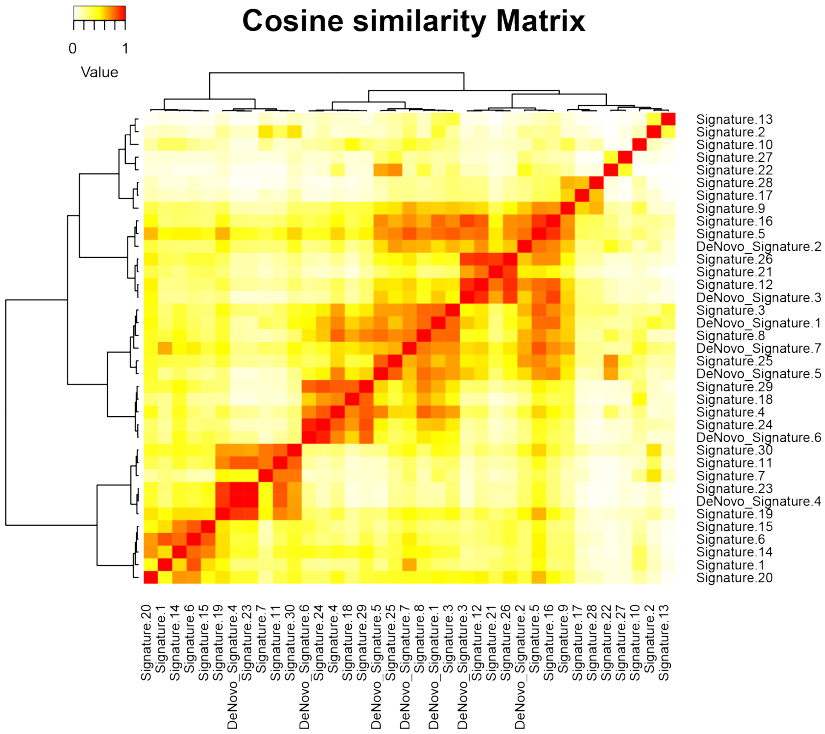

Figure 3: Cosine similarity heatmap. De novo extracted signatures are compared with previously described mutational signatures from COSMIC database. The color code represents the similarity (0 to 1) between each pair of signatures. Signatures are grouped by similarity using hierarchical clustering.

5.3 Estimating the exposures of mutational signatures

After extracting mutational signatures, Palimpsest will estimate the contribution of each signature to each individual tumor genome. Different graphical representations are also generated at this step, like the distribution of mutations according to the 96 mutation categories or the comparison of mutations occurring on the transcribed and non-transcribed strand.

```
-----------------------------------------------------------------
# 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.)
-----------------------------------------------------------------
```

To explore only known mutational signatures, users can start directly at this stage, providing as input signatures a set of previously described signature (e.g. from COSMIC databse) instead of de novo signatures. In this example, we use NMF to extract the contribution of 10 mutational signatures previously described in liver cancers.

```
-------------------------------------------------------------------
# Calculating contributions (exposures) of COSMIC signatures operative
  in liver cancers

# Selecting relevant signatures
>liver_signature_names <- c("Signature.1", "Signature.4", "Signature.5",
                            "Signature.17",Signature.22", "Signature.23",
                            "Signature.24")

>liver_signatures <- COSMIC_Signatures[liver_signature_names,]

# Calculating contributions (exposures) of known liver cancer 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.)
 -------------------------------------------------------------------
```

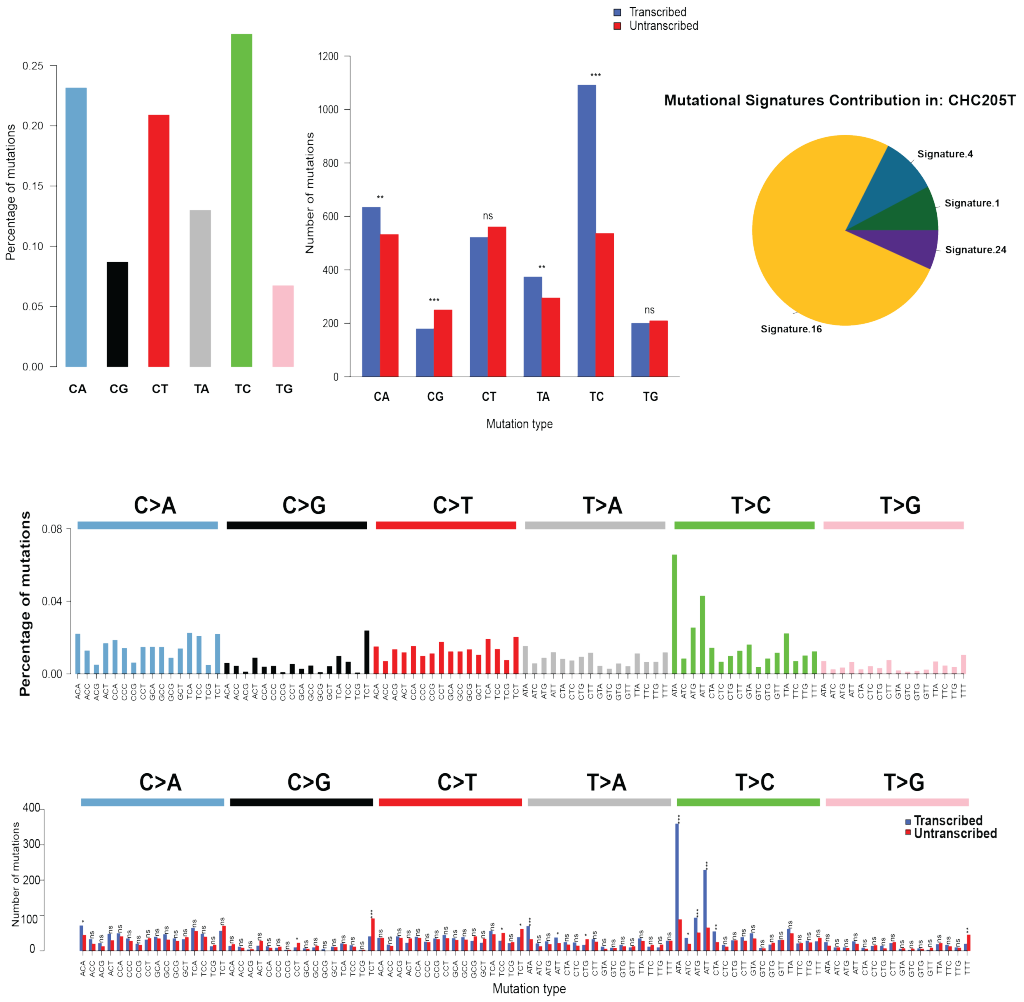

Figure 4: Fitting extracted mutational signatures in a tumor profile. The pie chart indicates the contribution of each mutational signature to the mutation catalogue of the tumor. The barplots indicate the distribution of mutations according to the 6 substitution types and the 96 mutation categories, distinguishing or not mutations occurring on the transcribed and non-transcribed strands.

The deconvolution_exposure function also allows to represent the contribution of signatures to each sample across the series.

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

>deconvolution_exposure(mutSign_nums = signatures_exp$sig_nums,
                        mutSign_props = signatures_exp$sig_props,
                        sig_cols = mycol)

-----------------------------------------------------------------
```

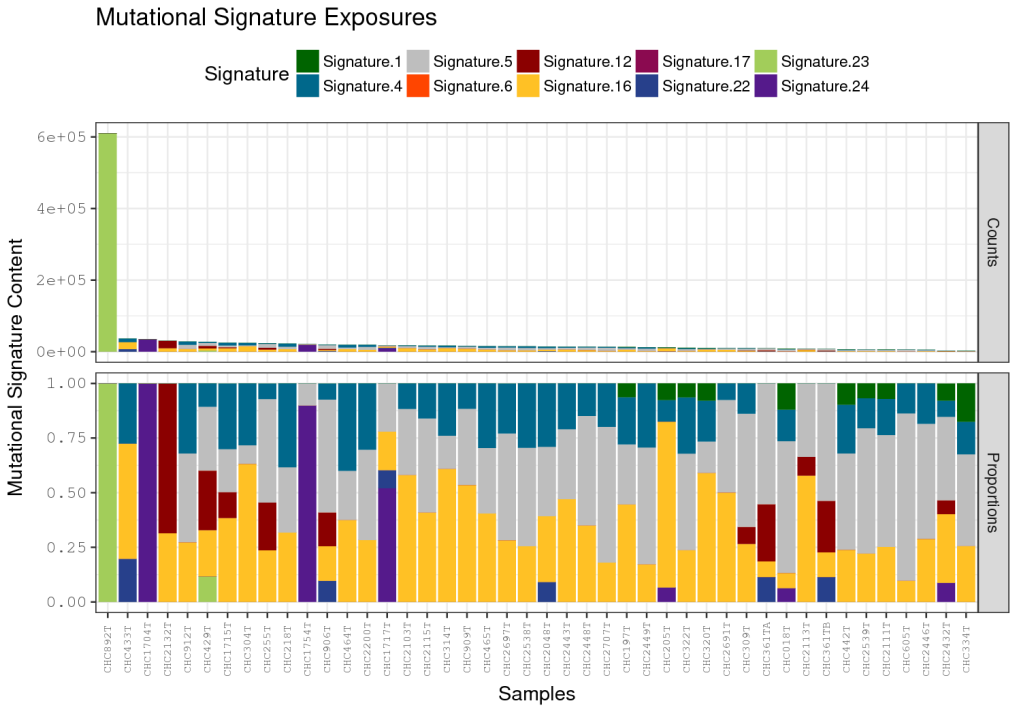

Figure 5: Contribution of mutational signatures across the tumor series. The number (top) and proportion (bottom) of mutations attributed to each signature in each tumor is represented with a color code, as illustrated in the figure legend.

5.4 Assigning the most likely signature at the origin of each mutation

An important question related to mutational signatures is to understand what mutational process gave rise to each individual mutation, in particular driver mutations. We developed a statistical framework to estimate the probability of each mutation being due to each process, considering the mutation category and the contribution of each signature to the corresponding tumor genome (Letouzé, Shinde et al., 2017). This method is implemented in the palimpsestOrigin function.

```
-----------------------------------------------------------------
>vcf <- palimpsestOrigin(vcf,
                         type = "SNV",
                         sample.col = "Sample",
                         mutcat.col = "mutcat3",
                         signature_contribution = signatures_exp$sig_nums,
                         input_signatures = liver_signatures)
-----------------------------------------------------------------
```

The palimpsestOrigin function adds columns to the vcf table giving the probability of each mutation being due to each process and the most likely causal process. For example, the TP53 R249S mutation in tumor CHC1754T is probably due to the aflatoxin B1-related signature 24 (probability = 0.99) whereas the TP53 R248L mutation in CHC2200T is most likely due to the tobacco-related signature 4 (probablility = 0.60).

```
##           Sample       Driver Sig.1.prob Sig.4.prob Sig.5.prob Sig.6.prob
## 147726  CHC2048T  TP53 splice          0     0.2703     0.0911          0
## 163094  CHC2200T TP53 p.R248L          0     0.5998     0.2601          0
## 299156  CHC2132T TP53 p.R337G          0     0.0000     0.0000          0
## 991607  CHC1704T TP53 p.R249S          0     0.0000     0.0000          0
## 1021113 CHC1754T TP53 p.R249S          0     0.0000     0.0094          0
## 1049273  CHC314T TP53 p.H179Y          0     0.2358     0.1625          0
##         Sig.12.prob Sig.16.prob Sig.17.prob Sig.22.prob Sig.23.prob
## 147726    0.0000000   0.1071000           0      0.5315           0
## 163094    0.0000000   0.1401000           0      0.0000           0
## 299156    0.6849648   0.3150352           0      0.0000           0
## 991607    0.0000000   0.0000000           0      0.0000           0
## 1021113   0.0000000   0.0000000           0      0.0000           0
## 1049273   0.0000000   0.6017000           0      0.0000           0
##         Sig.24.prob      Sig.max
## 147726       0.0000 Signature.22
## 163094       0.0000  Signature.4
## 299156       0.0000 Signature.12
## 991607       1.0000 Signature.24
## 1021113      0.9906 Signature.24
## 1049273      0.0000 Signature.16
```

We can then estimate the cumulative contribution of signatures to each driver gene in the cohort, and identify processes preferentially associated with mutations in specific driver genes. In this example, CTNNB1 mutations are preferentially associated with the alcohol-related signature 16.

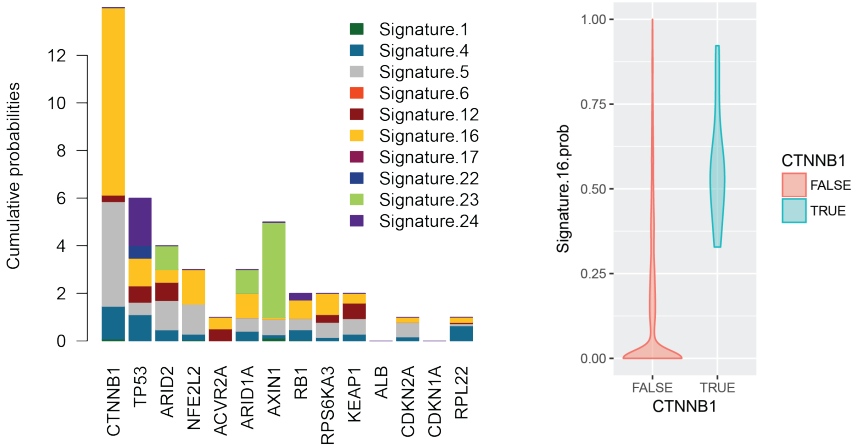

Figure 6: Association between mutational signatures and driver events in a cohort of 44 liver tumors. The barplot on the left shows the cumulative probabilities of driver gene mutations being due to each mutational process. The violin plot on the right compares the probabilities of being due to signature 16 between CTNNB1 mutations and other coding mutations, showing that CTNNB1 mutations are preferentially related to signature 16.

6 Clonality Analysis

Palimpsest provides functions to classify mutations as early clonal or late subclonal, and to monitor the evolution of mutational signatures along tumorigenesis in each tumor.

6.1 Cancer cell fraction estimation

First, the function cnaCCF_annot() allows to calculate the cancer cell fraction (CCF), i.e. the proportion of tumor cells harboring each mutation. This is done by adjusting the variant allele fraction (VAF) for the tumor purity (provided in the annotation file) and the absolute copy-number at each locus (provided in the CNA file). The 95% confidence interval of the CCF is also calculated, and mutations are classified as subclonal if the upper boundary of the 95% confidence interval is under a defined threshold (here, 0.95). The detailed formulas for CCF estimation are described in Letouzé, Shinde et.al 2017.

```
-----------------------------------------------------------------
>vcf <- cnaCCF_annot(vcf=vcf,
                     annot_data = annot_data,
                     cna_data = cna_data,
                     CCF_boundary = 0.95)
-----------------------------------------------------------------
```

The cnaCCF_annot() function adds several columns to the vcf file, including tumor purity, coverage log ratio (LogR), total number of copies at the locus (ntot), number of major (Nmaj) and minor (Nmin) alleles, cancer cell fraction (CCF) and confidence interval boundaries (CCF.min, CCF.max) and the assigned clonality status.

```
##            Sample Type CHROM       POS REF ALT Tumor_Varprop Gender   Purity
## 108542   CHC1715T  SNV  chr2 224505349   C   G     0.4090909      M 0.770000
## 108542.1 CHC1715T  SNV  chr2 224505349   C   G     0.4090909      M 0.770000
## 81516     CHC912T  SNV  chr3  38463087   A   C     0.3513514      M 0.790000
## 121547    CHC218T  SNV chr11  26573323   A   C     0.1890244      M 0.540000
## 19324    CHC361TA  SNV chr11 132378508   G   T     0.2521008      F 0.604701
## 112329   CHC1715T  SNV  chr4 180460585   T   A     0.3193277      M 0.770000
##               LogR     ntot Nmaj Nmin      nmut mult       CCF   CCF.adj
## 108542   0.6520767 1.995380    1    1 1.0606837    1 1.0606837 1.0000000
## 108542.1 0.6520767 1.995380    1    1 1.0606837    1 1.0606837 1.0000000
## 81516    0.5655972 2.107951    1    1 0.9274259    1 0.9274259 0.9274259
## 121547   1.1677274 2.944167    2    1 0.8785610    1 0.8785610 0.8785610
## 19324    0.8948178 2.111318    1    1 0.8618666    1 0.8618666 0.8618666
## 112329   0.7248928 1.996149    1    1 0.8281929    1 0.8281929 0.8281929
##            CCF.min  CCF.max Clonality
## 108542   0.7918133 1.345882    clonal
## 108542.1 0.7918133 1.345882    clonal
## 81516    0.6438099 1.243511    clonal
## 121547   0.6144208 1.196566    clonal
## 19324    0.6051267 1.162137    clonal
## 112329   0.6143125 1.066039    clonal
```

6.2 Clonality plots

The cnaCCF_plots() function then generates graphical outputs for each tumor to visualize and assess the definition of clonal and subclonal mutations.

First, a scatterplot representing the relationship between the variant allele fraction (VAF) of somatic mutations and the local coverage log-ratio illustrates the relationship between copy-number and VAF, and reveals a cloud of low VAF subclonal mutations. Second, histograms are generated representing the distribution of VAF and CCF across all the mutations of the tumor, with a color code to distinguish clonal (orange) from subclonal (blue) mutations.

```
----------------------------------------------------------------------
>cnaCCF_plots(vcf = vcf,
              resdir = resdir)
----------------------------------------------------------------------
```

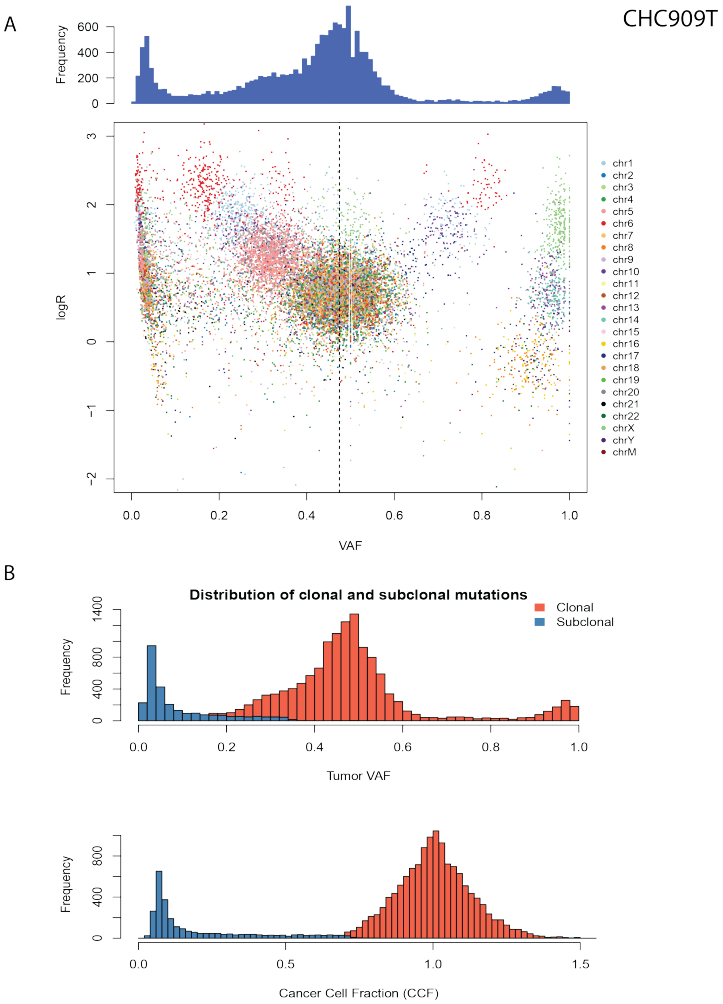

Figure 7: Genome-wide distribution of variant allele fractions (VAF) and cancer cell fractions (CCF) in a tumor. The scatterplot on the top shows the relationship between the VAF of mutations and local copy-number. Each point represents a somatic mutation, colored according to the chromosome, with the VAF on the x axis and coverage log-ratio between

tumor and normal on the y axis. Deletions lead to a decreased log-ratio and increased VAF. Duplications lead to an increased log-ratio and a decreased (for mutations on the non-duplicated chromosome) or increased (for mutations on the duplicated chromosome) VAF. Subclonal mutations are visible as a cloud of mutations with very low VAF not explained by copy-number changes. The bottom histogram represents the distribution of VAF and CCF across all mutations in the tumor, with a color code indicating their clonal (orange) or subclonal (blue) status.

6.3 Temporal evolution of mutational signatures

Once the clonality of each mutation has been established, Palimpsest allows to analyze the evolution of mutational signatures between early clonal and late subclonal mutations. First, we deconvolute the contribution of each signature to each tumor, considering clonal and subclonal mutations separately.

```
-------------------------------------------------------------------
# Estimate the contribution of each signature to clonal mutations in each tumor
>vcf.clonal <- vcf[which(vcf$Clonality=="clonal"),]
>propMutsByCat.clonal <- palimpsestInput(vcf = vcf.clonal,
                                         type="SNV",
                                         sample.col = "Sample",
                                         mutcat.col = "mutcat3",
                                         proportion = TRUE)
>signatures_exp_clonal <- deconvolution_fit(vcf = vcf.clonal,
                                            type = "SNV",
                                            input_data = propMutsByCat.clonal,
                                            threshold = 6,
                                            input_signatures = liver_signatures,
                                            sig_cols = mycol,
                                            plot = F,
                                            resdir = resdir.)

# Estimate the contribution of each signature to subclonal mutations in each tumor
>vcf.subclonal <- vcf[which(vcf$Clonality=="subclonal"),]
>propMutsByCat.subclonal <- palimpsestInput(vcf = vcf.subclonal,
                                            type = "SNV",
                                            sample.col = "Sample",
                                            mutcat.col = "mutcat3",
                                            proportion = TRUE)
>signatures_exp_subclonal <- deconvolution_fit(vcf = vcf.subclonal,
                                               type = "SNV",
                                               input_data = propMutsByCat.subclonal,
                                               threshold = 6,
                                               input_signatures = liver_signatures,
                                               sig_cols = mycol,
                                               plot = F,
                                               resdir = resdir.)
-------------------------------------------------------------------
```

Then, we use the palimpsest_DissectSigs() function to generate visual representations of the 96 mutation category spectrums in early and late mutations and to compare the proportions of early and late mutations attributed to each signature in each tumor.

```
-------------------------------------------------------------------
# Generate tumor-wise comparisons of clonal and subclonal mutations
>palimpsest_DissectSigs(vcf = vcf,
                        signatures_exp_clonal = signatures_exp_clonal,
                        signatures_exp_subclonal = signatures_exp_subclonal,
                        sig_cols = mycol,
                        resdir = resdir)
-------------------------------------------------------------------
```

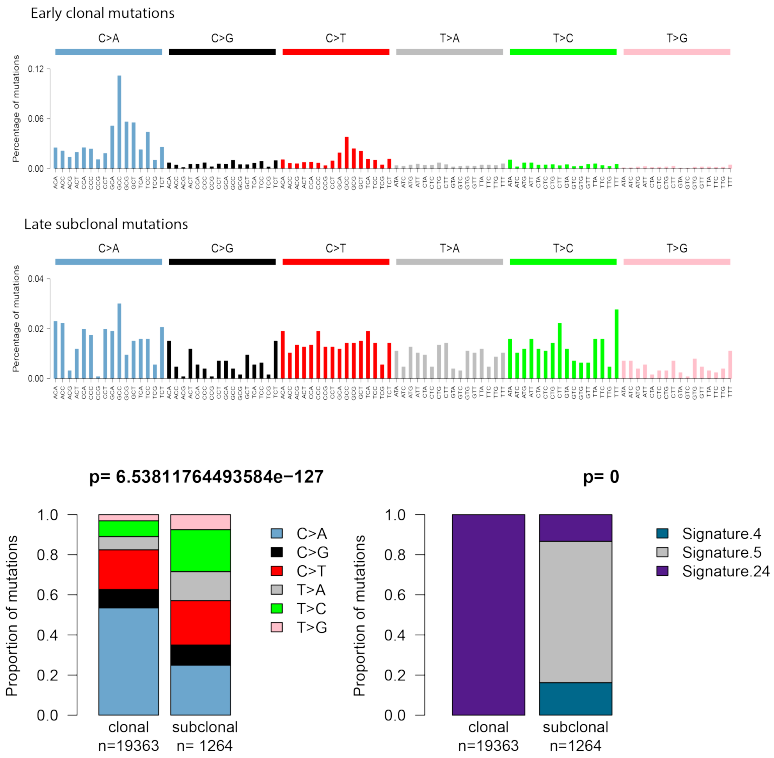

Figure 8: Temporal evolution of mutational signatures in a liver tumor. The top panel shows the 96 mutation category spectrums of clonal (top) and subclonal (bottom) mutations. The bottom left panel shows the proportions of the 6 substitution types in clonal and subclonal mutations, with the p-value above (chi-square test). The bottom right panel shows the proportions of clonal and subclonal mutations attributed to each signature, with the p-value above (chi-square test).

The evolution of mutational signatures between clonal and subclonal mutations across the series can also be conveniently visualized using the following function:

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

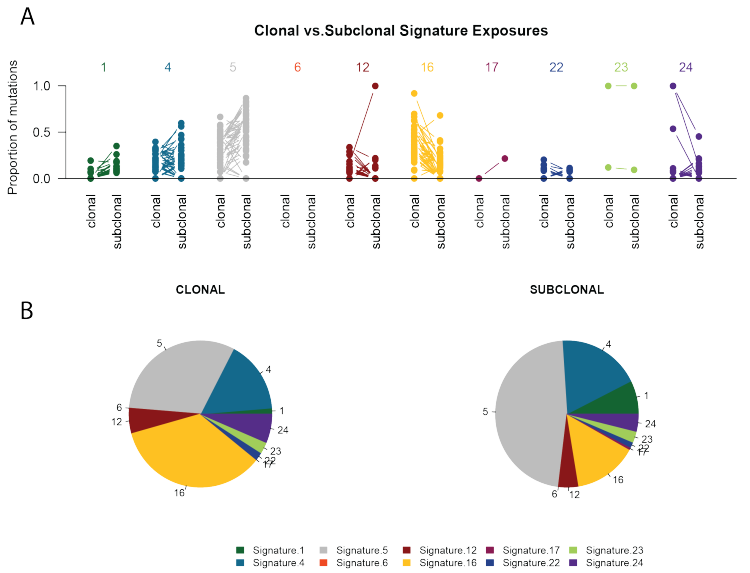

Figure 9: Evolution of mutational signatures between early clonal and late subclonal mutations in a series of tumors. Top: proportion of mutations attributed to each mutational signature in the clonal and subclonal mutations of each tumor (connected with a line). Clon. clonal; Sub. subclonal. Bottom: Average contribution of each signature to clonal and subclonal mutations in our series of liver tumors.

6.4 Timing chromosomal gains

Several tools exist to identify early clonal and late subclonal copy-number alterations from next-generation sequencing data. In addition, Palimpsest allows to time the occurrence of chromosome duplications in molecular time. When a chromosome is duplicated (e.g. from 2 to 3 copies), mutations harbored by the duplicated chromosome copy are also duplicated and have an increased VAF. By contrast, mutations harbored by the other copy or acquired after the duplication are just present in one of the 3 copies and have a lower VAF. As a result, early duplications have few duplicated mutations compared to late duplications. Palimpsest uses the number of duplicated and non-duplicated mutations to estimate the timing of each chromosome duplication (see Letouzé, Shinde et al. 2017 for more detailed methodology). A cytoband table is required for graphical representations. Cytoband tables in hg19 and hg38 formats are provided with the package.

```
-----------------------------------------------------------------------
# Annotate vcf with chromomal gain timings
>chrom_dup_time <- chrTime_annot(vcf = vcf,
                                 cna_data = cna_data,
                                 cyto = cyto)

# Visualizing timing plots
chrTime_plot(vcf = vcf,
             point.mut.time = point.mut.time,
             resdir = resdir)
-----------------------------------------------------------------------
```

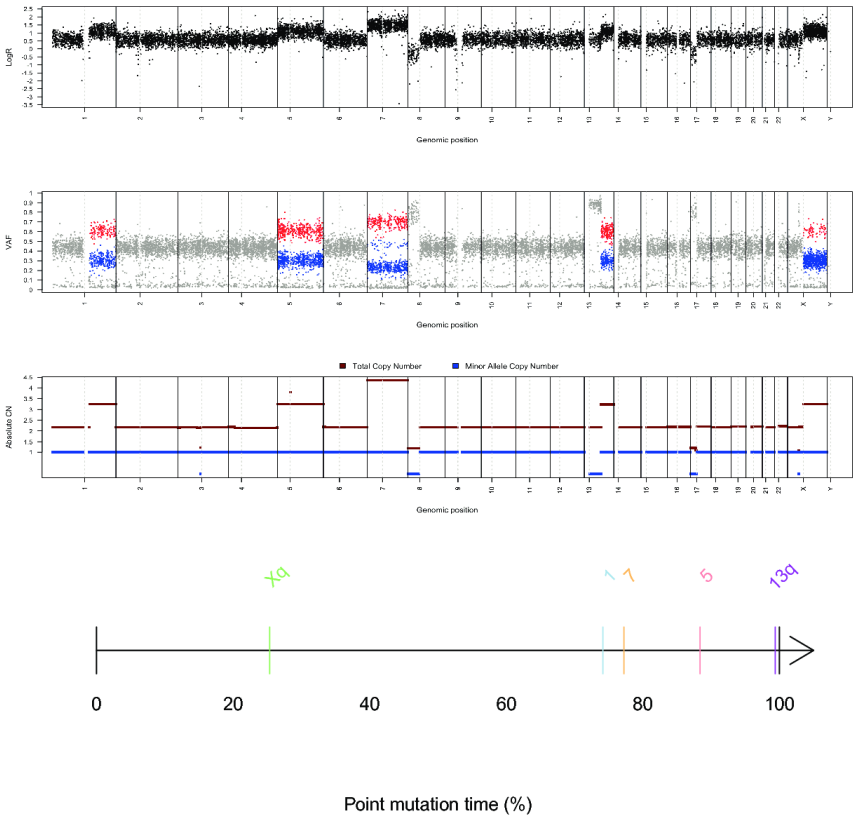

Figure 10: Chromosome duplication timing in a liver tumor. Top: The log-ratio (top) and variant allele fraction (VAF, middle) of each somatic mutation are represented throughout the genome, with the absolute copy-number below. For each duplicated chromosome region, the color of the points on the VAF graph distinguishes duplicated (red) from non-duplicated (blue) somatic mutations. Here, most duplications have a similar amount of duplicated/non-duplicated mutations, indicating that they occurred late, when most mutations were already present. By contrast, the duplication of chromosome X has a lower number of duplicated mutations, so this duplication occurred earlier. Bottom: The timing of chromosome duplications is represented in point mutation time (0 = time when no mutation had been acquired yet, 100 = time when they had all been acquired).

7 Structural Variant (SV) Signatures:

Palimpsest implements an adaptation of the mutational signature analysis framework to structural variants (SVs), as initially described by Nik-Zainal et al. (Nature 2016) and modified by Letouzé, Shinde et al. (Nat Commun 2017). SVs are first classified into 38 categories considering the type (deletion, tandem duplication, inversion, inter-chromosomal translocation), size (<1kb, 1-10kb, 10-100kb, 100kb-1Mb, 1-10Mb, >10Mb) and clustered nature of rearrangements. The same statistical tools as those used for point mutations are then applied to extract SV signatures and their contribution to each tumor. Palimpsest also provides graphical representations of tumor SV profiles as CIRCOS plots and barplots showing the number of events per SV category. The workflow (below) is very similar to the workflow used for point mutation analysis.

```
-----------------------------------------------------------------
>library(bedr);library(RCircos) # libraries necessary for SV analysis
# Preprocessing SV inputs
sv.vcf <- preprocessInput_sv(input_data = sv_data,
                             ensgene = ensgene,
                             resdir = resdir)
propSVsByCat <- palimpsestInput(vcf = sv.vcf,
                                sample.col = "Sample",
                                type = "SV",
                                mutcat.col = "Category",
                                proportion = FALSE)

# Running de novo signature analysis
denovo_signatures <- deconvolution_nmf(input_data = propSVsByCat,
                                       type = "SV",
                                       range_of_sigs = 2:12,
                                       nrun = 20,
                                       method = "brunet",
                                       resdir = resdir)
-----------------------------------------------------------------
```

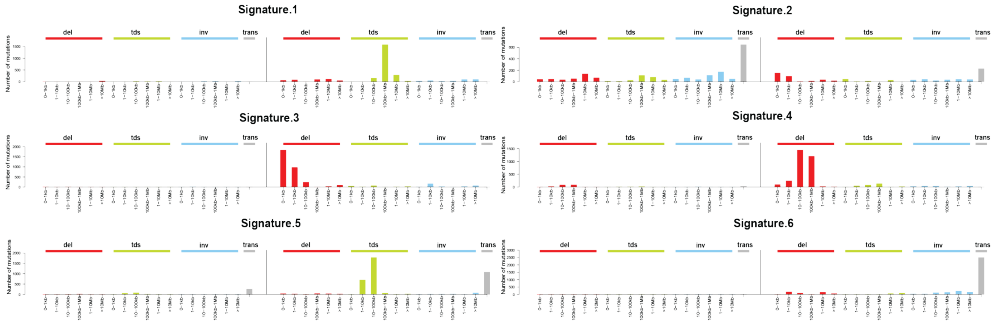

Figure 11: Six rearrangement signatures identified in our series of liver tumors. Structural rearrangements were classified in 38 categories considering their type (del: deletion, dup: tandem duplication, inv: inversion, trans: interchromosomal translocation) and size, and distinguishing clustered from non-clustered events. The probability of each rearrangement category in each signature is represented, with rearrangement types indicated above and rearrangement sizes below.

```
-----------------------------------------------------------------
# Calculating contributions(exposures) of signatures in each sample
SVsignatures_exp <- deconvolution_fit(vcf = sv.vcf,
                                      type = "SV",
                                      input_data = propSVsByCat,
                                      threshold = 6,
                                      input_signatures = denovo_signatures,
                                      sig_cols = mycol.sv,
                                      plot = T,
                                      resdir = resdir)

# Plotting the exposures of signatures across the series
deconvolution_exposure(SVsignatures_exp$sig_nums,
                       SVsignatures_exp$sig_props,
                       sig_cols = mycol.sv)

# Estimate the probability of each event being due to each process
sv.vcf <- palimpsestOrigin(vcf=sv.vcf,
                           type = "SV",
                           sample.col = "Sample",
                           mutcat.col = "Category",
                           signature_contribution=SVsignatures_exp$sig_nums,
                           input_signatures=denovo_signatures)
-----------------------------------------------------------------
```

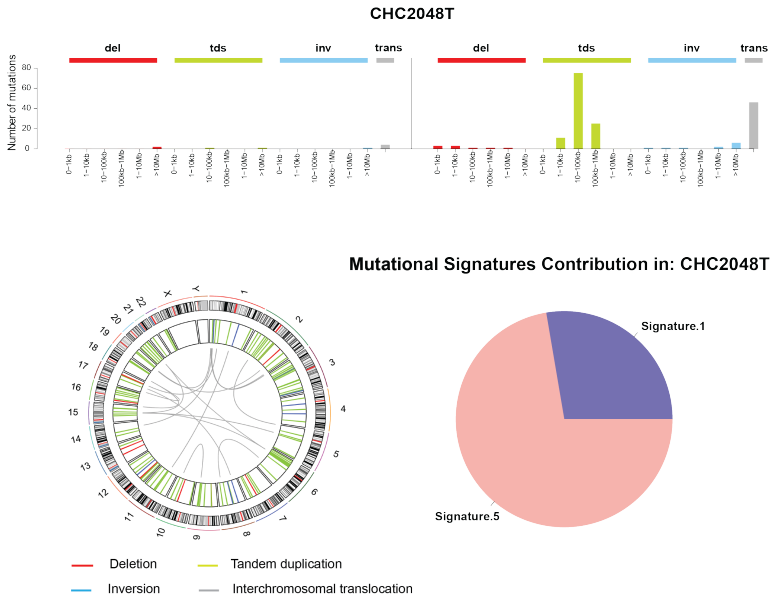

Figure 12: Fitting extracted structural variant signatures in a tumor profile. The barplots indicate the distribution of mutations according to the 38 structural rearrangement categories. CIRCOS plot represents the structural rearrangement profile of the tumor. The pie chart indicates the contribution of each mutational signature to the mutation catalogue of the tumor.

8 Natural history of tumors

The last step in a typical Palimpsest analysis is to generate a schematic representation of the natural history of each tumor. The palimpsest_plotTumorHistories() function generates tumor history plots indicating the number of clonal and subclonal mutations identified, the contribution of mutational signatures to early clonal and late subclonal mutations, the timing of chromosome duplications and the timing of driver mutations and SVs (manually annotated by the user). Driver events are also colored according to the the mutational process that most likely generated them.

```
-----------------------------------------------------------------
>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.)
-----------------------------------------------------------------
```

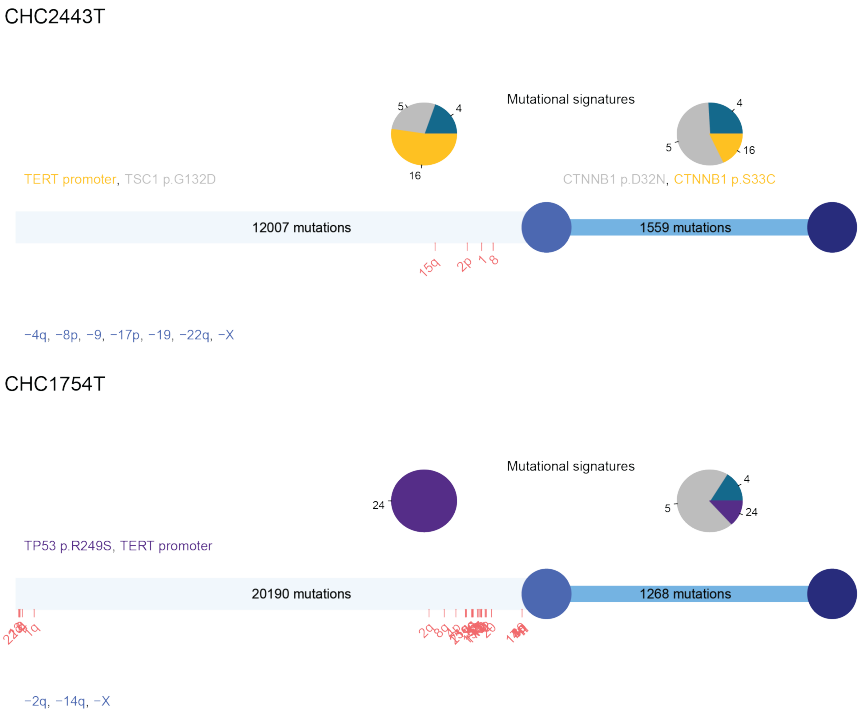

Figure 13: Tumor history plots representing the natural history of two liver tumors. The middle blue circle represents the last common ancestor of all tumor cells in the sample and the dark blue circle represents the final tumor sample. In the first tumor (alcohol-related, top), mutational signature 16 was the predominant mutational process in early clonal mutations whereas signature 5 increased in late tumor development. This tumor displays

clonal mutations of TERT promoter and TSC1, and two subclonal mutations of CTNNB1 attributed to signatures 5 and 16. In the second tumor (aflatoxin B1-related, below), highly active mutational signature 24 generated >20,000 clonal mutations including TP53 and TERT promoter mutations. This signature was less active in subclonal events, again replaced by signature 5. Deletions of ALB and AXIN1 genes also occurred early in tumor development, and synchronous acquisition of multiple gains occurred right before subclonal diversification of the tumor sample.

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_1.1.7   rmarkdown_1.8   knitr_1.19
## [13] xfun_0.1        stringr_1.3.0   digest_0.6.15   evaluate_0.10.1
```