SATS:
Signature Analyzer for Targeted Sequencing

User Guide

DongHyuk Lee[1], William Wheeler[2] and Bin Zhu[3]

[1] Department of Statistics, Pusan National University, Busan, Korea

[2] Information Management Services (IMS), Inc. Rockville, MD, USA

[3] Division of Cancer Epidemiology and Genetics, National Cancer Institute, National Institutes of Health, Bethesda, MD, USA

Version 0.0.7

September 12, 2023

Contents

1 Introduction

For the targeted sequenced tumors, applying the existing mutational signature analysis tools is challenging because they are developed for whole-exome or whole-genome sequencing (WES or WGS). Since the sequenced genomic sizes and contexts of targeted panels differ from WES or WGS, not all mutational signatures detected by WES or WGS can be found using the targeted sequencing panels. On the other hand, panel-based signatures may include *de novo* presence of signatures. To identify mutational signatures accurately, we developed a new method, SATS (Signature Analyzer for Targeted Sequencing), for targeted sequenced tumors. In this guide, we present a step-by-step guide to detecting mutational signatures in targeted sequenced tumors, using the **R** package *SATS*.

The SATS procedure comprises four steps, which encompass the detection of *de novo* signatures, mapping the *de novo* signatures to the reference TMB signatures, the estimation of signature activities, and the calculation of reference signature expectancies. In cases where a sufficiently large number of tumors are sequenced, the complete set of SATS steps can be effectively deployed (as detailed in Section 4). Conversely, in situations where access to only a limited number of tumors, or even a single tumor, is possible, the signature refitting steps involving the estimation of signature activities and expectancies remain applicable (as discussed in Section 5).

2 Installation

To install from Github directly, one can use the **R** package *devtools*:

```
if (!requireNamespace("devtools", quietly = TRUE))
        install.packages("devtools")
devtools::install_github("binzhulab/SATS/source")
```

Alternatively, SATS_0.0.7.tar.gz (for Unix) or SATS_0.0.7.zip (for Windows, R version >= 4.1) from the Github page[1] can be installed using the following commands:

```
install.packages("SATS_0.0.7.tar.gz", repos = NULL, type = "source")
install.packages("SATS_0.0.7.zip", repos = NULL, type = "win.binary")
```

Once installed, *SATS* can be loaded on **R** by calling

```
> library(SATS)
```

3 Example data

For illustrative purposes, we simulated a 96 $\times$ 10027 mutation catalog matrix **V**, which contains 10027 targeted sequenced tumors for 96 single base substitution (SBS) types. Each element of the matrix is generated from the Poisson distribution (`rpois()` function) with the mean corresponding to each element of $\mathbf{L} \circ \mathbf{W}_T\mathbf{H}_T$, where $\circ$ is elementwise multiplication, **L** is a given panel size matrix with the same 96 $\times$ 10027 size of **V**, $\mathbf{W}_T$ is the profile matrix of the tumor mutation burden (TMB, in the unit of the number of mutations per million base pairs) based signatures of size 96$\times$6 for 6 SBS signatures (SBS1, SBS2/13, SBS4, SBS5, SBS40, SBS89) and $\mathbf{H}_T$ is the signature activity matrix of size 6$\times$10027. All 4 matrices are stored in `SimData` with corresponding names: **V** (`SimData$V`), **L** (`SimData$L`), $\mathbf{H}_T$ (`SimData$TrueH`), $\mathbf{W}_T$ (`SimData$TrueW_TMB`).

```
> data(SimData, package = "SATS")
> dim(SimData$V)
[1] 96 10027
> SimData$V[1:6, 1:6]
        Sample1 Sample2 Sample3 Sample4 Sample5 Sample6
A[C>A]A       0       0       0       0       0       0
A[C>A]C       0       0       0       0       0       1
A[C>A]G       0       0       0       0       0       0
```

[1] https://github.com/binzhulab/SATS

```
A[C>A]T       0       0       0       0       0       0
C[C>A]A       2       0       0       0       1       0
C[C>A]C       1       0       0       0       0       0
```

```
> dim(SimData$L)
[1] 96 10027
> SimData$L[1:6, 1:6]
        Sample1 Sample2 Sample3 Sample4 Sample5 Sample6
A[C>A]A  0.0511  0.0511  0.1014  0.0445  0.0511  0.1014
A[C>A]C  0.0390  0.0390  0.0745  0.0338  0.0390  0.0745
A[C>A]G  0.0131  0.0131  0.0236  0.0115  0.0131  0.0236
A[C>A]T  0.0424  0.0424  0.0848  0.0371  0.0424  0.0848
C[C>A]A  0.0624  0.0624  0.1150  0.0539  0.0624  0.1150
C[C>A]C  0.0515  0.0515  0.0911  0.0435  0.0515  0.0911
```

```
> dim(SimData$TrueH)
[1]     6 10027
> SimData$TrueH[, 1:5]
         Sample1  Sample2  Sample3  Sample4  Sample5
SBS1    7.31e+01 4.57e-96 1.24e-75 5.69e-83 2.95e-07
SBS2_13 1.80e-19 4.13e-47 2.05e-60 2.19e+01 2.74e+01
SBS4    1.90e+02 1.12e+02 1.39e-21 2.23e-40 5.48e-42
SBS5    2.70e+02 3.07e-11 2.80e+01 6.57e+01 1.01e+02
SBS40   3.61e-01 1.44e+01 1.14e-06 1.20e-14 1.58e-11
SBS89   7.54e+00 1.78e-08 4.93e-17 1.83e-19 6.01e-13
```

```
> dim(SimData$TrueW_TMB)
[1] 96  6
> head(SimData$TrueW_TMB)
            SBS1  SBS2_13   SBS4     SBS5  SBS40  SBS89
A[C>A]A 1.21e-04 6.84e-04 0.0236 7.22e-03 0.0215 0.0214
A[C>A]C 5.40e-04 5.64e-04 0.0322 9.81e-03 0.0176 0.0203
A[C>A]G 1.94e-04 9.48e-04 0.0697 8.88e-03 0.0178 0.0512
A[C>A]T 2.19e-04 2.09e-04 0.0206 4.97e-03 0.0141 0.0172
C[C>A]A 4.66e-05 6.58e-04 0.0492 4.87e-03 0.0173 0.0178
C[C>A]C 3.73e-04 6.08e-04 0.0679 5.62e-03 0.0170 0.0147
```

4 SATS procedures

The primary purpose of the SATS algorithm is to conduct mutational signature analysis for targeted sequenced tumors, summarized in Figure 1. Somatic mutations (e.g., single base substitutions [SBS]) detected by the targeted sequencing are summarized into the mutation type matrix **V** based on the mutation and its genomic context. Given the mutation type matrix **V**, *de novo* tumor mutation burden (TMB) signatures are discovered by SigneR (Rosales et al., 2017), adjusting the panel sizes (Section 4.2). Next, we map the *de novo* signature obtained by signeR to the reference SBS TMB signatures (found in `RefTMB$SBS_W`; Section 5) using the penalized non-negative least squares (Tibshirani, 2011; Section 4.3). Then given the mutation type matrix (**V**), panel size of each tumor (**L**) and the mapped reference TMB signature profiles, signature activities are estimated for every tumor by the Expectation-Maximization (EM) algorithm simultaneously (Section 4.4). Finally, we computed the expected number of mutations due to each signature, called signature expectancy, for all tumors simultaneously (Section 4.5).

4.1 Input data

For the mutation catalog matrix (**V**), it could be an **R** `dataframe` or `matrix` of which elements are non-negative counts. Each column of **V** corresponds to a tumor (or sample), while its row represents a mutation type. Although the selection of the number of signatures is independent of the order of mutation type, we specified the order of mutation type according to the COSMIC database (SBS signature[2]).

The element of the panel size matrix (**L**) represents the length of the trinucleotide contexts per million base pair to the corresponding sequencing panel. Similarly, each column of **L** corresponds to a tumor (or sample) while its row represents a mutation type.

[2] `https://cancer.sanger.ac.uk/signatures/sbs/`

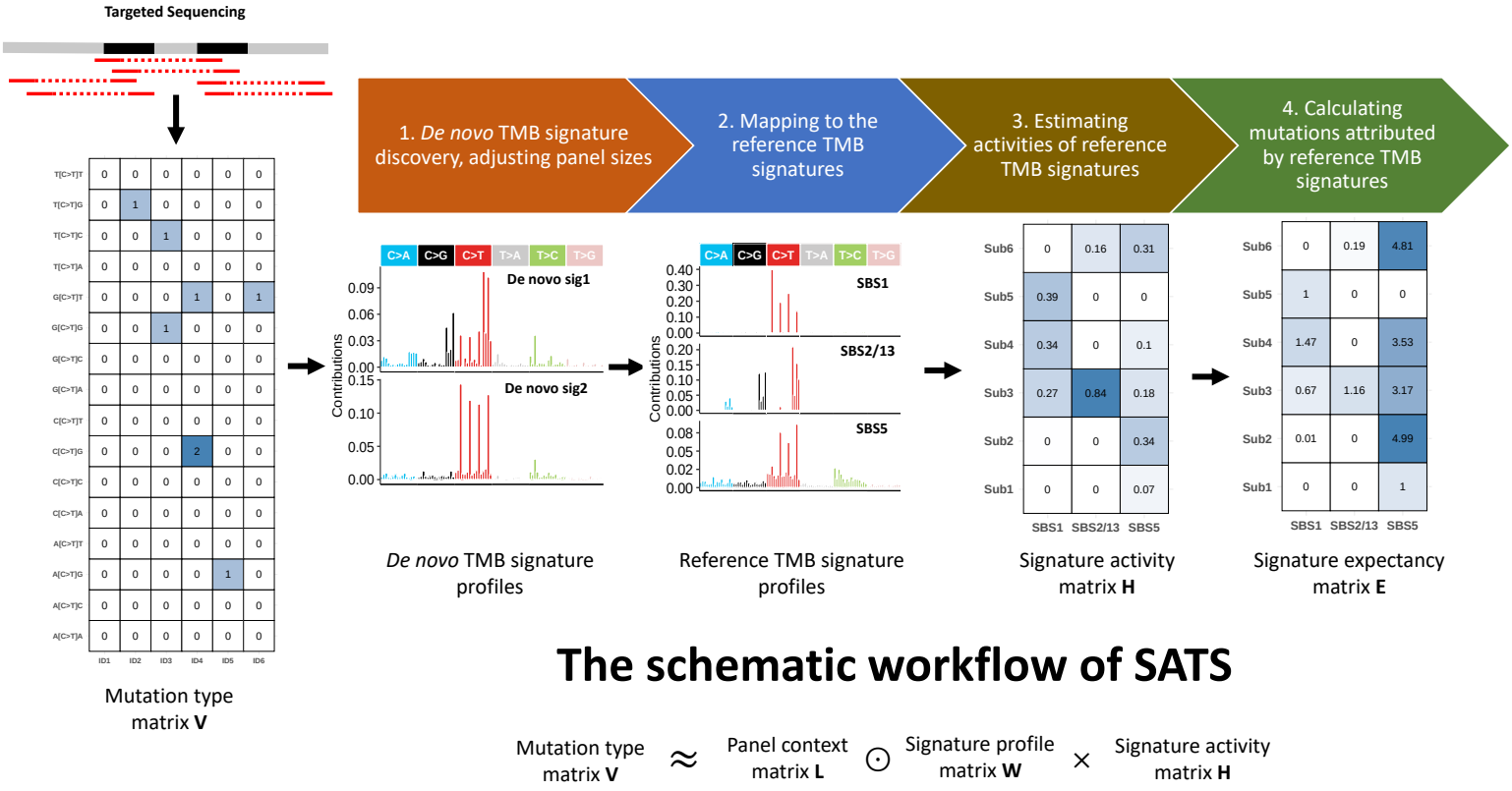

Figure 1: The schematic workflow of SATS (Signature Analyzer for Targeted Sequencing).

4.2 Application of the signeR algorithm

To estimate *de novo* SBS signatures $\widehat{\mathbf{W}}$, we first apply signeR (need to reference) which finds an optimal solution $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{H}}$ such that $\mathbf{V} \approx \mathbf{L} \circ \widehat{\mathbf{W}}\widehat{\mathbf{H}}$. Since signeR uses a Bayesian approach with Markov chain Monte Carlo (MCMC), it is computationally intensive and time-consuming if the entire 96×10027 matrices $\mathbf{V}$ and $\mathbf{L}$ are used.

Therefore, we used pooled tumor of 100 combined samples, noting that pooling would not change the TMB-based signature profile. We are first interested in estimating $\mathbf{W}$. In the codes below, `V_sum` and `L_sum` are combined mutation count matrix and the sum of panel size for these corresponding combined samples used to create `V_sum`, respectively. Then they are main arguments, `M` and `Opport`

of `signeR()` function as shown below. Note that the resulting $\widehat{\mathbf{H}}$ has a different dimension from the original matrix **V** by pooling samples.

Another argument of `signeR()` function we used is `nlim` which specifies the range of possible number of optimal signatures. It needs much time to run the function if the upper bound of `nlim` is large. From our limited experience, the optimal number of signatures computed by `signeR()`, ranging between 2 to 5. Once `signeR()` is done, `BICboxplot()` can be used to visualize the number of optimal signatures by signeR (Figure 2).

```
> library(tidyverse)
> V_group <- as.data.frame(t(SimData$V))
> V_group$srting <- as.character((1:dim(V_group)[1])%/%100)
> V_sum <- V_group %>% group_by(srting) %>%
        dplyr::select('A[C>A]A':'T[T>G]T')%>%summarise_all(sum)
Adding missing grouping variables: `srting`
> V_sum=as.data.frame(V_sum[,-1])
> rownames(V_sum) = paste0("G",rownames(V_sum))
>
> L_group <- as.data.frame(t(SimData$L))
> L_group$srting <- as.character((1:dim(L_group)[1])%/%100)
> L_sum <- L_group %>% group_by(srting) %>%
        dplyr::select('A[C>A]A':'T[T>G]T')%>%summarise_all(sum)
Adding missing grouping variables: `srting`
> L_sum=as.data.frame(L_sum[,-1])
> rownames(L_sum) = paste0("G",rownames(L_sum))
> ## Extract W hat
> library(signeR)
> set.seed(4022022)
> signeR_re <- signeR(M=V_sum, Opport=L_sum, nlim=c(1,5))
Evaluating models with the number of signatures ranging from 1 to 5,
please be patient.
Evaluating 1 signatures.
EM algorithm:
|==========================================================| 100%
Running  Gibbs sampler for 1 signature...Done.
Evaluating 2 signatures.
EM algorithm:
|==========================================================| 100%
```

```
Running  Gibbs sampler for 2 signatures...Done.
Evaluating 3 signatures.
EM algorithm:
|============================================================| 100%
Running  Gibbs sampler for 3 signatures...Done.
Evaluating 4 signatures.
EM algorithm:
|============================================================| 100%
Running  Gibbs sampler for 4 signatures...Done.
Evaluating 5 signatures.
EM algorithm:
|============================================================| 100%
Running  Gibbs sampler for 5 signatures...Done.
The optimal number of signatures is 2.
Running  Gibbs sampler for 2 signatures...Done.
>BICboxplot(signeR_re)
```

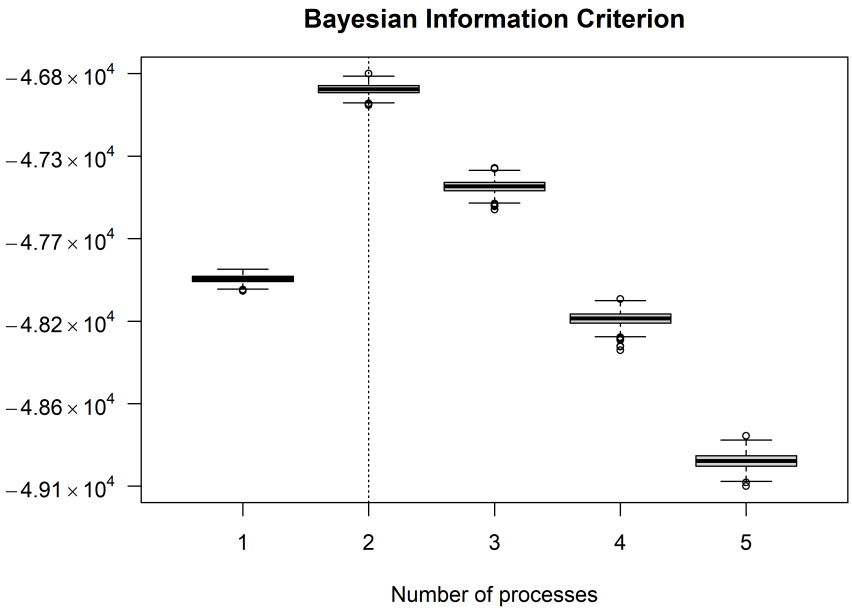

Figure 2: BIC (Bayesian Information Criterion) plot of signeR

4.3 Mapping *de novo* TMB-based signatures to reference TMB signatures

Because the targeted sequencing panels have limited somatic mutations compared with WGS or WES, signeR identifies a few *de novo* TMB-based signatures (adjusted by panel size matrix **L**; Figure 3), which could be a linear combination of reference TMB signatures. Therefore we could consider mapping *de novo* signeR signature matrix $\widehat{\mathbf{W}}$ to reference TMB signature $\mathbf{W}_0$ (stored in `RefTMB$SBS_W`).

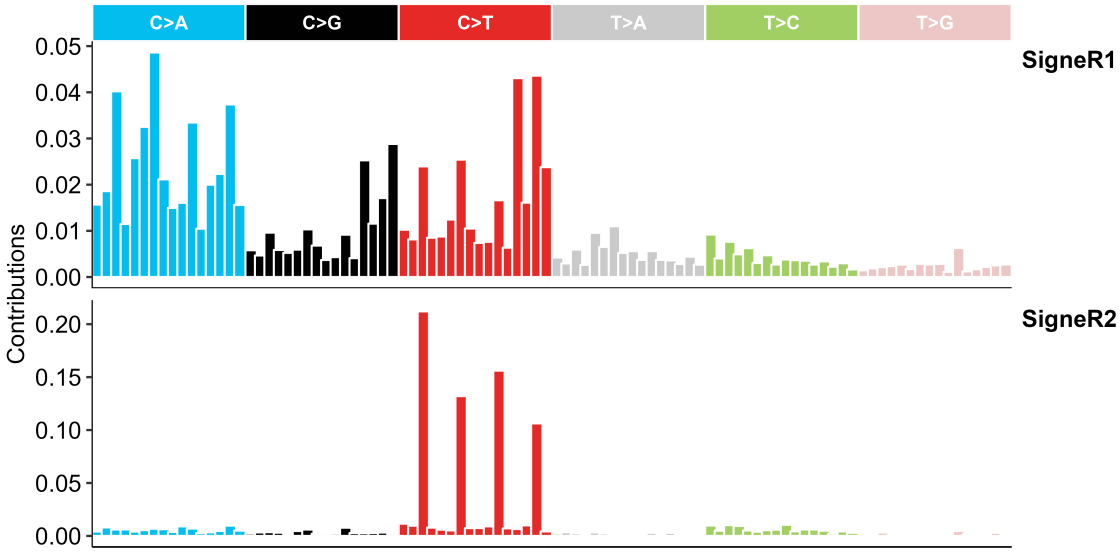

Figure 3: *De novo* TMB signature profiles identified by signeR (Section 4.2)

To this end, we applied penalized non-negative least squares (pNNLS) for selecting the reference TMB signatures. Specifically, we repeated pNNLS 100 times to reduce the randomness of cross-validation involved in pNNLS. Then reference TMB signatures are selected with a coefficient greater than 0.1 in more than 80 repeats.

It can be done by calling `MappingSignature(W_hat, W_ref=NULL, niter=100, cutoff.I2=0.1, min.repeats=80)` in the *SATS* package. `W_hat` is a *de novo* signatures from signeR (or any other signature analysis tool). `W_ref` is for the reference signature profiles, which will be mapped to. And the default is `NULL` with `RefTMB$SBS_W` used. The remaining `nIter`, `cutoff.I2` and `min.repeats`

specifies the number of pNNLS repetitions (the default is 100), cutoff coefficient value to select signatures (the default is 0.1) and the minimum number of repetitions greater than cutoff coefficient value (the default is 80), respectively.

As a result, the mapped reference TMB signatures are SBS1, SBS2/13, SBS4, SBS5, SBS40, and SBS89 with 100 frequencies. Their profiles are drawn in Figure 4. They are the same as signatures used to generate data (`colnames(SimData$TrueW_TMB)`).

```
> W_hat <- signeR_re$Phat
> MappedSig <- MappingSignature(W_hat = W_hat, W_ref = RefTMB$SBS_W)
> MappedSig
  Reference freq
1      SBS1  100
2   SBS2_13  100
3      SBS4  100
4     SBS40  100
5      SBS5  100
6     SBS89  100
> colnames(SimData$TrueW_TMB)
[1] "SBS1"    "SBS2_13" "SBS4"    "SBS5"    "SBS40"   "SBS89"
```

4.4 Estimation of signature activities by an expectation-maximization algorithm

Let $\mathbf{W}^*$ be the mapped reference TMB signature profiles. Then given the original mutation catalog matrix **V**, the panel size matrix **L**, and $\mathbf{W}^*$, we can extract the signature activities $\widehat{\mathbf{H}}$ with the function `EstimateSigActivity(V, L, W, n.start=50, iter.max=5000, eps=1e-5)` in the *SATS* package. The three main input matrices are **V**, **L**, and $\mathbf{W}^*$ (or any signature profiles).

Because the EM algorithm is used to estimate signature activities, `n.start`, `iter.max` and `eps` control EM part. Because the convergence to a local saddle point can be an issue of the EM algorithm, it would be good practice to try multiple initial values (`n.start`, the default is 50). For each initial value, the default value

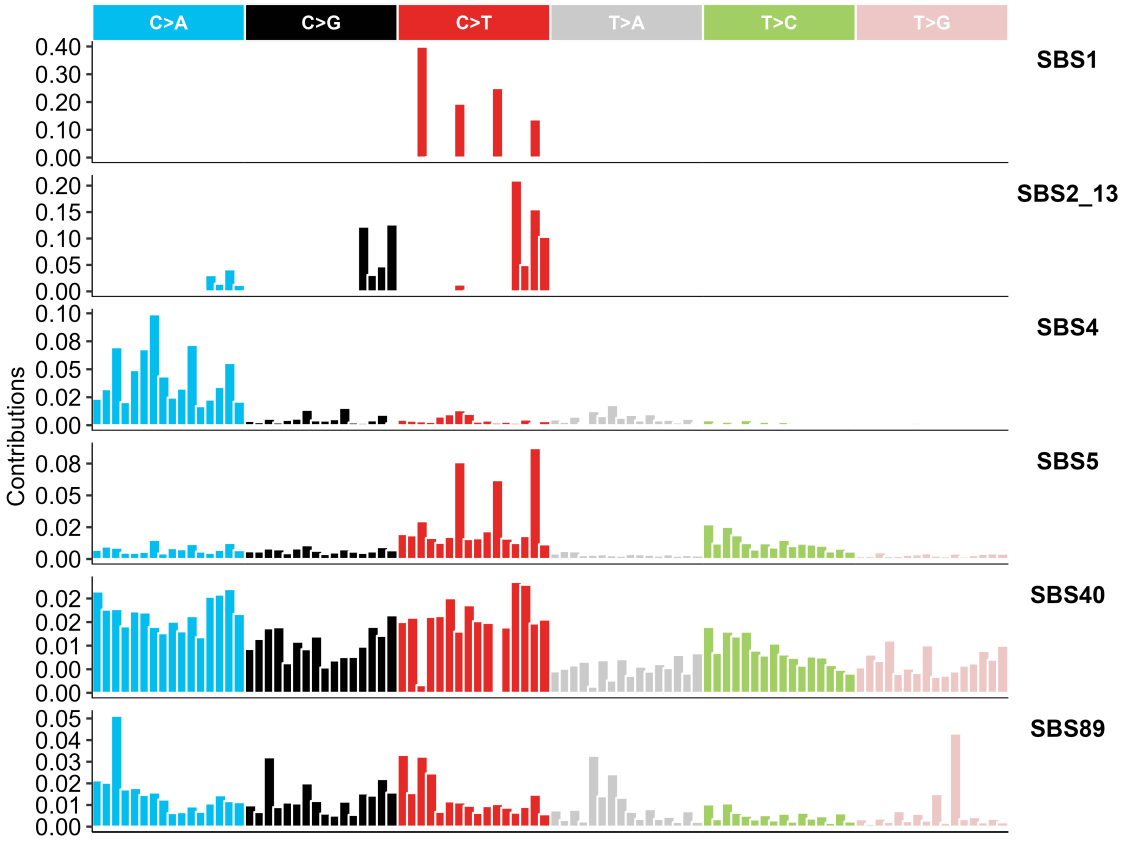

Figure 4: The mapped reference SBS TMB signatures obatined by `MappingSignature()` function

of the maximal iteration of the EM algorithm (`iter.max`) is 5000 and the stopping tolerance (`eps`) is set to 10^{-5}.

The function `EstimateSigActivity()` returns the signature activities matrix estimate $\widehat{\mathbf{H}}$, convergence status (1 for converged and 0 for not converged), and the largest log likelihood among `n.start` initial values.

```
> library(gtools)
> SBS.list <- mixedsort(unique(MappedSig[,"Reference"]))
> W_star <- as.matrix(RefTMB$SBS_W[,SBS.list])
> H_hat <- EstimateSigActivity(V = SimData$V, L = SimData$L, W = W_star)
> round(H_hat$H[, 1:5], 2)
       Sample1 Sample2 Sample3 Sample4 Sample5
```

```
SBS1     295.42     0.0     0.0     0.0    0.00
SBS2_13    0.00     0.0    11.7    28.3   18.40
SBS4     180.96     0.0     0.0     0.0   20.52
SBS5       4.22     0.0     0.0    93.3    3.03
SBS40     44.64    85.5     0.0    21.7   72.39
SBS89      0.00    52.6     0.0     0.0    0.00
> H_hat$converged
[1] 1
> H_hat$loglike
[1] -0.2172882
```

4.5 Calculation of signature expectancy

The final step is to compute the expected number of mutations attributed by reference TMB signatures (i.e., signature expectancy) given matrices **L**, $\mathbf{W}^*$ (the mapped reference TMB signature profiles; see Section 4.3) and $\widehat{\mathbf{H}}$ (the signature activities matrix estimate; see Section 4.4). The signature expectancy can be computed via `CalculateSigExpectancy(L, W, H)` in the *SATS* package. It decomposes mutations of a tumor into mutations due to the identified signatures. For example, Sample 1 has 14 mutations, about 4.8 caused by signature 1, 7 by signature 4, about 0.2 by signature 5 and two by signature 40.

```
> SigExp <- CalculateSigExpectancy(L = SimData$L, W = W_star, H = H_hat$H)
> round(SigExp[,1:5], 2)
        Sample1 Sample2 Sample3 Sample4 Sample5
SBS1       4.79    0.00       0    0.00    0.00
SBS2_13    0.00    0.00       1    1.08    0.82
SBS4       7.04    0.00       0    0.00    0.77
SBS5       0.21    0.00       0    2.93    0.07
SBS40      1.95    3.87       0    0.99    3.34
SBS89      0.00    2.13       0    0.00    0.00
> colSums(SimData$V[,1:5])
Sample1 Sample2 Sample3 Sample4 Sample5
     14       6       1       5       5
```

5 Signature refitting of SATS

An additional benefit of the SATS algorithm is its capability to estimate signature activities and expectancies, even when working with a single tumor sample, as long as the set of signatures specific to a particular cancer type is known. It becomes possible to estimate their signature activities and expectancies by utilizing the pan-cancer repertoire of SBS and DBS TMB signatures. A detailed illustrative example is provided in Section 5.2 for clarification

5.1 The pan-cancer repertoire of reference signatures

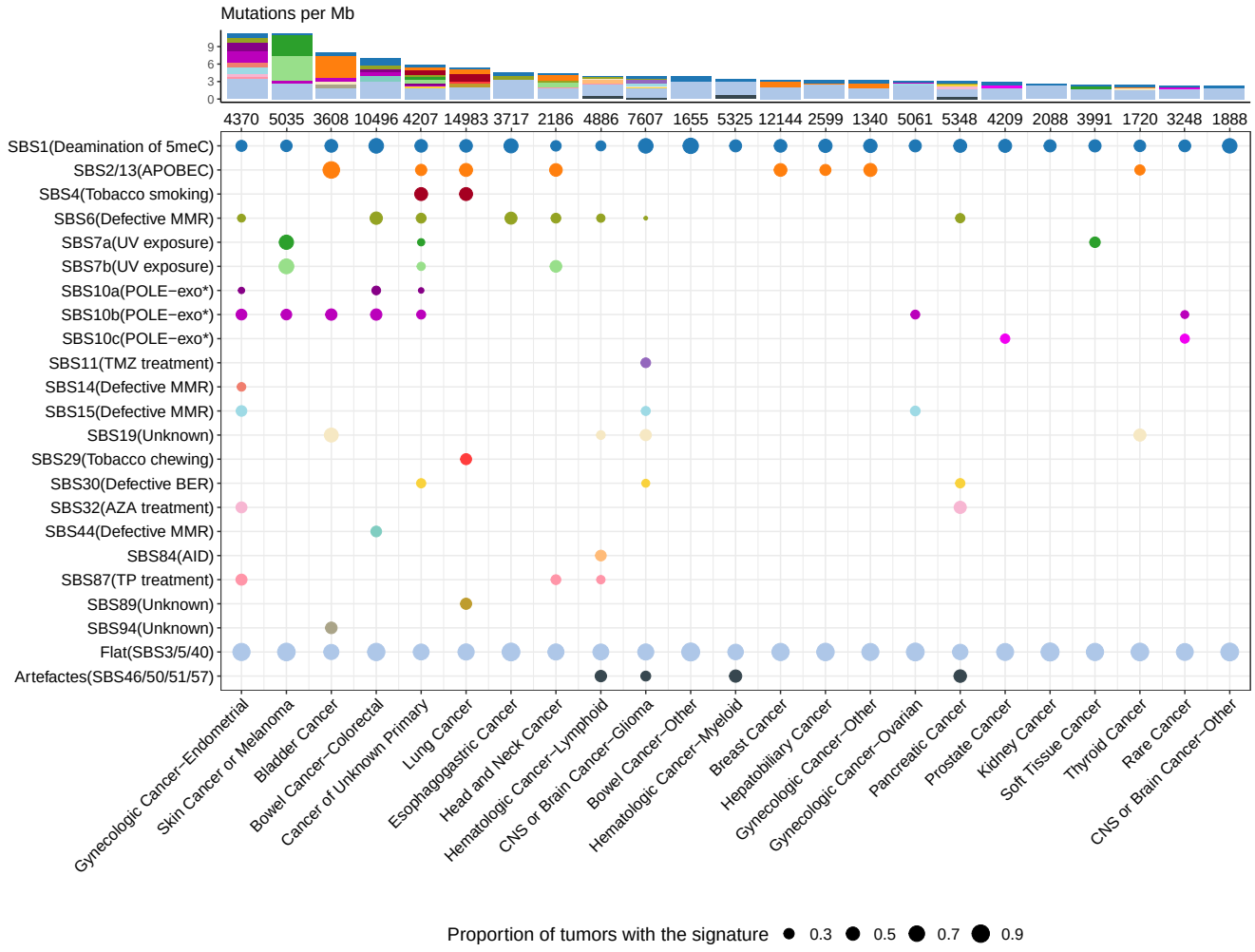

Figure 5: Repertoire of SBS mutational signatures in the AACR Project GENIE

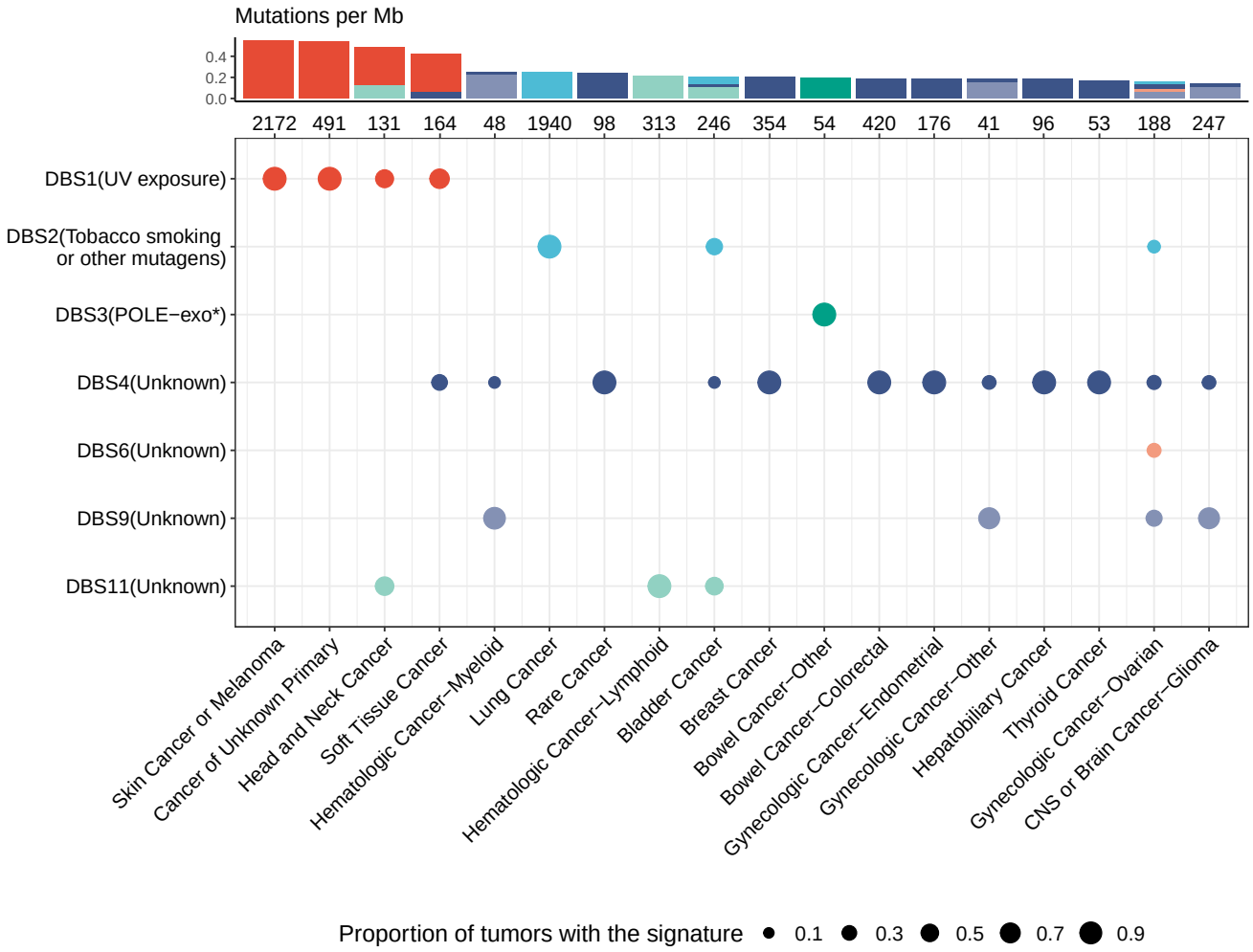

Figure 6: Repertoire of DBS mutational signatures in the AACR Project GENIE

In most practical applications, medical practitioners usually have only a few or even just a single tumor sample for a specific caner type. While it is not feasible to derive *de novo* signatures from these limited set of tumors, it is possible to obtain estimates of mutational activities for the tumor in question. This can be achieved by utilizing predefined reference TMB signatures through the refitting process. To this end, a pan-cancer repertoire of mutational signatures applied to the targeted sequenced tumors from the AACR Project GENIE is obtained by SATS procedure and presented in Figures 5 and 6.

The COSIMC reference TMB signature profiles and cancer specific signature

information for both SBS and DBS are stored in `RefTMB`: `RefTMB$SBS_W` and `RefTMB$SBS_refSigs` contain the SBS TMB signature profiles of size 96 by 76 and the list of cancer specific SBS signature names, respectively. Similarly, `RefTMB$DBS_W` and `RefTMB$DBS_refSigs` contain the COSMIC DBS TMB signature and the list of cancer specific DBS signature names.

```
> data(RefTMB, package = "SATS")
> dim(RefTMB$SBS_W)
[1] 96 76
> RefTMB$SBS_W[1:6, 1:6]
            SBS1   SBS2_13    SBS3    SBS4      SBS5      SBS6
A[C>A]A 1.21e-04 6.84e-04 0.0155 0.0236 7.22e-03 9.37e-05
A[C>A]C 5.40e-04 5.64e-04 0.0213 0.0322 9.81e-03 1.99e-04
A[C>A]G 1.94e-04 9.48e-04 0.0104 0.0697 8.88e-03 9.14e-05
A[C>A]T 2.19e-04 2.09e-04 0.0114 0.0206 4.97e-03 4.96e-05
C[C>A]A 4.66e-05 6.58e-04 0.0183 0.0492 4.87e-03 4.37e-04
C[C>A]C 3.73e-04 6.08e-04 0.0287 0.0679 5.62e-03 1.37e-03
> head(RefTMB$SBS_refSigs, 8)
                cancerType  COSMIC
1          Bladder Cancer    SBS1
2          Bladder Cancer  SBS10b
3          Bladder Cancer   SBS19
4          Bladder Cancer SBS2_13
5          Bladder Cancer    SBS5
6          Bladder Cancer   SBS94
7 Bowel Cancer-Colorectal    SBS1
8 Bowel Cancer-Colorectal  SBS10a
```

These resources can be readily applied to the refitting process on a daily basis in the clinic as elaborated in the subsequent section.

5.2 An analysis of a single tumor with the reference TMB signatures

The basic procedure follows a similar structure as detailed in the previous sections (Sections 4.4, 4.5 and 4.6). However, the mapped reference TMB signature profiles, $\mathbf{W}^*$ (Sections 4.2 and 4.3), are not accessible due to the limited sample size. In such cases, we can substitute the mapped reference TMB signatures with those provided

in Figure 5, given that the cancer type of the particular tumor sample is known. For example, if a targeted sequenced tumor is obtained from breast cancer, then SBS1, SBS2_13, SBS5, and SBS40 can be taken from the list of cancer specific SBS signature names `RefTMB$SBS_refSigs` as illustrated below. The signature repertoire from other cancer types is available by simply replacing `"Breast Cancer"` with the appropriate cancer type. A list of available cancer types can be found `RefTMB$SBS_refSigs[, "cancerType"]`.

```
> unique(RefTMB$SBS_refSigs[, "cancerType"])
[1] "Bladder Cancer"                "Bowel Cancer-Colorectal"
[3] "Bowel Cancer-Other"            "Breast Cancer"
[5] "Cancer of Unknown Primary"     "CNS or Brain Cancer-Glioma"
[7] "CNS or Brain Cancer-Other"     "Esophagogastric Cancer"
[9] "Gynecologic Cancer-Endometrial" "Gynecologic Cancer-Other"
[11] "Gynecologic Cancer-Ovarian"    "Head and Neck Cancer"
[13] "Hematologic Cancer-Lymphoid"   "Hematologic Cancer-Myeloid"
[15] "Hepatobiliary Cancer"          "Kidney Cancer"
[17] "Lung Cancer"                   "Pancreatic Cancer"
[19] "Prostate Cancer"               "Rare Cancer"
[21] "Skin Cancer or Melanoma"       "Soft Tissue Cancer"
[23] "Thyroid Cancer"
> RefTMB$SBS_refSigs[RefTMB$SBS_refSigs$cancerType == "Breast Cancer", ]
       cancerType  COSMIC
15 Breast Cancer    SBS1
16 Breast Cancer SBS2_13
17 Breast Cancer   SBS40
18 Breast Cancer    SBS5
```

For the sake of illustration, we employ a single simulated tumor derived from a skin Cancer which is stored in `SimData$SingleTumorEx`. This example dataset compromises mutation counts corresponding to 96 SBS categories (in the `singleV` column) and the associated sequencing context (in the `singleL` column). In this example, we provide a comprehensive walkthrough of the entire process using SBS counts. However, it's important to note that DBS mutation counts can be analyzed in a similar manner, leveraging the resources available in `RefTMB$DBS_W` and `RefTMB$DBS_refSigs`.

```
> head(SimData$SingleTumorEx)
        singleV  singleL
A[C>A]A       0 0.051058
A[C>A]C       1 0.039010
A[C>A]G       0 0.013127
A[C>A]T       0 0.042381
C[C>A]A       0 0.062427
C[C>A]C       0 0.051509
> RefTMB$SBS_refSigs[RefTMB$SBS_refSigs$cancerType == "Skin Cancer or Melanoma", ]
                cancerType COSMIC
110 Skin Cancer or Melanoma   SBS1
111 Skin Cancer or Melanoma SBS10b
112 Skin Cancer or Melanoma   SBS5
113 Skin Cancer or Melanoma  SBS7a
114 Skin Cancer or Melanoma  SBS7b
```

Subsequently, with signatures SBS1, SBS5, SBS7a, SBS7b, and SBS10b, which have been identified within a pan-cancer repertoire (as demonstrated in the code above), we can proceed to estimate mutational activities and signature expectancies as outlined follows.

In the subsequent code, the `SBS.list` constructs a new $\mathbf{W}^*$, which will serve as an input for the `CalculateSigExpectancy(L, W, H)` function. The estimated activity, denoted as `H_hat$H`, is then employed to calculate the signature expectancy using the `CalculateSigExpectancy(L, W, H)` function.

```
> ## W star from a pan-cancer repertoire
> SBS.list <- RefTMB$SBS_refSigs[RefTMB$SBS_refSigs$cancerType ==
                  "Skin Cancer or Melanoma", "COSMIC"]
> SBS.list
[1] "SBS1"   "SBS10b" "SBS5"   "SBS7a"  "SBS7b"
> W_star <- as.matrix(SimData$W_TMB[,SBS.list])
>
> ## Estimate activity
> V1 <- SimData$SingleTumorEx[, 1, drop = FALSE]
> L1 <- SimData$SingleTumorEx[, 2, drop = FALSE]
> H_hat <- EstimateSigActivity(V = V1, L = L1, W = W_star)
> H_hat$H
        singleV
```

```
SBS1    47.35763
SBS10b 322.13271
SBS5   234.46156
SBS7a  199.14687
SBS7b  311.48489
> H_hat$converged
[1] 1
> H_hat$loglike
[1] -0.3019976
>
> ## Estimate Expectancy
> SigExp <- CalculateSigExpectancy(L = L1, W = W_star, H = H_hat$H)
> round(SigExp, 2)
        singleL
SBS1       0.77
SBS10b     5.86
SBS5       8.81
SBS7a      7.52
SBS7b     12.05
> sum(V1)
[1] 35
> sum(SigExp)
[1] 35
```

References

Rosales, R. A., Drummond, R. D., Valieris, R., Dias-Neto, E. and da Silva, I. T. (2017). signeR: an empirical Bayesian approach to mutational signature discovery. *Bioinformatics* **33**, 8–16.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society, Series B* **73**, 273–282.