Signature Fit Multi-Step (FitMS) user manual

signature.tools.lib version: 2.0
latest edit: 22/10/2021

Andrea Degasperi, University of Cambridge, UK
ad923@cam.ac.uk

1. Introduction

This document describes how to use *FitMS* to estimate the presence of mutational signatures in whole genome sequencing data of cancer samples.

Mutational signatures are patterns of somatic mutations, that are caused by endogenous or exogenous mutational processes. Since the introduction of mutational signature analysis in 2012, many mutational signatures have been discovered.

Just in the last few years (2017-2021), three major projects of cancer whole genome sequencing have been completed: PCAWG, approximately 3000 cancer genomes collected from a number of independent previous studies; Hartwig, approximately 3500 metastatic cancer whole genomes; and Genomics England, approximately 15000 cancer genomes from the UK national health service.

In our recent mutational signature analysis of these three datasets, we identified 82 high quality single base substitution signatures (SBS), and showed that each organ has its own set of organ-specific signatures. Most importantly, we showed that in each organ the majority of cancers contain a set of common signatures, while a small subset, around 5-10%, contain one or two additional rare signatures.

These results suggest that a good strategy for signature fitting would be to proceed with two steps: in the first step, the exposures of the common, organ specific signatures are estimated; in the second step, the algorithm attempts to find whether additional rare signatures are present. It should be noted, that the second step can be implemented in several ways, and we here provide four different implementations.

FitMS has been designed to be easy to pick up and use, while also giving the expert user a lot of different options to play with. For example, a user can specify the organ of origin of his/her samples, and the appropriate common and rare mutational signatures will be used. At the same time, an expert user can provide his/her own sets of common and rare signatures.

1.1 Signature Fit Concepts

Each mutational signature can be described mathematically as a vector $s \in \mathbb{R}_{\geq 0}^{m}$ such that the sum of all the values in the vector is equal to 1, i.e., $\sum_{i=1}^{m} s_i = 1$. Each value s_i indicates the likelihood of a certain mutation class to occur, given a corresponding underlying mutational process. For example, in cancers from heavy smokers, mutations of the C>A class are highly likely.

The value m is the number of mutation classes, also referred to as *channels*, considered in an analysis. For example, for single base substitution mutational signatures (SBSs), we typically consider 96 channels, defined as substitutions, converted to have a pyrimidine reference base, in their trinucleotide context (e.g., C[C>A]T is a C>A in a CCT context).

For each tumour sample, we can count the somatic mutations that belong to each mutation class, and thus obtain a vector $c \in \mathbb{N}_{\geq 0}^{m}$, called mutational *catalogue*. Each sample presents a mutational burden that is likely the result of multiple mutational processes that induced multiple

DNA lesions over time. Thus, we assume that c is a combination of a given set of k mutational signatures. If we represent the set of mutational signatures as a matrix S with dimensions $m \times k$, then we can write $c \approx Se$, with $e \in \mathbb{N}_{\geq 0}^{k}$ the vector of exposures, which indicates how many mutations in c are associated with each one of the k mutational signatures.

The procedure used to estimate the exposures e when S and c are given is known as *Signature fitting.* There are several algorithms and tools available to perform signature fit, each of them attempting to resolve issues and limitations with the procedure. A major issue is the continued discovery of new mutational signatures, which makes it impossible to simply use all of them at once in a matrix S. It is natural that as k is approaching m, there may be many vectors e that satisfy $c \approx Se$ equally well, leading to non-robust solutions that may even be biologically implausible.

Thus, recent efforts have focused on both trying to estimate the robustness or variability of the solution e, and to automatically select the subset of signatures that are more confidently present in a catalogue c.

The selection of a subset of k signatures to use for the matrix S can be the result of an automated procedure, for example leaving out signatures that appear to be not essential, or alternatively prior knowledge can be used, for example considering only mutational signatures that have been previously observed in a certain tumour type.

1.2 FitMS

In our recent work (*Degasperi et al. 2021*), we have brought evidence for three new concepts about mutational signatures, which can be exploited for improving signature fit. The first concept is the tissue-specificity of mutational signatures, which implies that each tissue type may have a slightly different version of mutational signature for each mutational process, as well as mutational signatures that are unique to that tumour type. The second concept is the distinction between common and rare signatures, which implies signature fit can be thought of a two-step process, where in the first step one can estimate the exposure of common tissue-specific signatures, while in the second step one can attempt to identify additional rare signatures. The third concept is the abundance of the common and rare signatures in a tissue-type or an organ. The number of common tissue-specific signatures in an organ is usually between 5 and 10, which is quite low with respect to the total of 82 signatures we reported in our study, and implies that solutions e are likely to be relatively robust if only these common signatures are used for fitting. Moreover, the proportion of samples that present rare signatures is only about 5-10%, and if a sample presents rare signatures, there is usually just one rare signature. This implies that after fitting the common signatures, one can just seek one additional signature that may improve the solution, without trying to fit several rare signatures at once.

These new concepts led to the definition of FitMS, which operates exactly as described above. Once again, in brief, given a certain tumour-type, in a first step, common tissue-specific signatures are fitted, and in a second step, one additional rare signature is search and considered if it contributes significantly to the solution.

In practice, there are different possible choices for the set of common and rare signatures to use. Here, when possible, we use the common signatures extracted from the Genomics England (GEL) dataset, as this dataset typically presents the largest number of samples for each organ, while if a certain tumour type is not present in GEL, we use the signatures we extracted from the ICGC/PCAWG data. For rare signatures we use a tier system, where the user can choose between Tier 1 (T1) or Tier 2 (T2) rare signatures. T1 rare signatures are those that were observed in a given organ in our study, while T2 rare signatures include all rare signatures found across all organs. In

general, T2 is preferable, since there may be signatures that occur in a specific organ but were not seen just because of the rarity of the signature or the low number of samples available for that organ.

The second step of FitMS, where one attempts to find a rare signature, can be implemented in several ways. While in (*Degasperi at al. 2021*) we showed two possible approaches, there are four different strategies implemented in this R package. The strategies are:

1. *constrainedFit*: we compute the residual $r = c - Se$ after fitting the common signatures using a constrained non-negative least squares algorithm that forces the residual to be mostly positive. Then we seek the rare signature that is most similar to the residual;
2. *partialNMF*: we compute the residual using a 'partial' non-negative matrix factorisation (NMF) approach, were we fit the common signatures while extracting one additional signature. Then we seek the rare signature that is most similar to the residual;
3. *errorReduction*: we fit the common signatures and then for all rare signatures we fit the common + one rare. We compare the errors and select the rare signatures that induce the highest error reduction compared to common alone;
4. *cossimIncrease*: similar to point 3, but instead of considering the reduction in error we consider the increase in cosine similarity to the catalogue.

We compared the accuracy of signature assignment of the four strategies using a simulation study and found that errorReduction is the best strategy on average. Expert users can adjust several parameters to experiment with the above methods and fine tune the outcome. All FitMS parameters are described in the documentation section below.

FitMS attempts to find rare signatures that improve upon the solution obtained with common signatures alone. It is also possible that multiple rare signatures can improve the solution of a sample, implying that there can be multiple competing valid candidate solutions. When this happens, FitMS selects the solution that leads to the highest improvement and reports it in its overall summary. In any case, the result object returned by FitMS contains all candidate solutions, and an expert user can compare the plots of all the candidate solutions (using the `plotFitMS` function). It is then possible to manually specify a preferred solution using the R function `fitMerge`.

Users can set the maximum number of rare signatures that are expected in a sample. While in most cases we expect no rare signatures and seldom one rare signature, there may be cases in which more than one rare signature could be present. FitMS will search for as many rare signatures as requested, as long as adding rare signatures improves the solution, so it is possible that the reported solutions use less rare signatures than the number requested. It should be noted that while more than one rare signature can be allowed, signatures are always added sequentially and checked one at a time starting from an existing valid solution. So, for example, if signature A improves on an initial solution, in order for signature A and B to be considered together, signature B needs to improve on the solution that includes A, or, alternatively, B needs to improve on the initial solution and A needs to improve on a solution that includes B.

To estimate the robustness of the solution and the uncertainty of the exposure estimates, bootstrap signature fit can be enabled. When using bootstrap, the mutational catalogue c is resampled multiple times using the relative proportion of each element of c as probability, which essentially perturbs the input and allows to estimate a distribution of the exposures. Point estimates are also computed as the median of each element of e.

After an initial fit has been computed, it is good practice to remove mutational signatures that contribute very little to the solution and that may be a consequence of overfitting. This *exposure filter* procedure is necessary to reduce the false positive assignment of signatures. However, this filter should be used with care since removing signatures that are truly present can also increase

false negatives. We provide two types of exposure filter. The first type is the widely used *fixed threshold*, where exposures are set to 0 if they are lower than a certain % of the total mutations in sample (here 5% by default). If bootstrap is used, we set the point estimate of the exposure to zero when more than 5% of the exposure distribution is lower than the threshold. While the fixed threshold works well most of the times, we considered that while some signatures require a relatively low number of mutations to be identified (typically if they are characterised by only a few classes of mutations, such as APOBEC signatures SBS2 and SBS13), other signatures require many mutations (typically if they are featureless where most mutation classes are involved, such as signatures SBS3 and SBS5). Thus, we implemented and introduce here a second type exposure filter, which is based on the *Gini coefficient* of a signature. Using the Gini-based exposure filter, each signature has a different threshold, depending on the shape of the mutational signature, thus allowing to detect distinctive signatures using a relatively low number of mutations, while requiring a relatively large number of mutations to detect featureless signatures.

Additional options that we provide with FitMS are parallelisation and the possibility to specify a random seed for when bootstrap is used.

2. Installation

FitMS is included as an R function in the signature.tools.lib R package. Thus, in order to use FitMS, one is required to install signature.tools.lib, which is available on GitHub:

https://github.com/Nik-Zainal-Group/signature.tools.lib

3. R function overview and documentation

The following information can be displayed in R typing `?FitMS`.

```
FitMS(catalogues, organ = NULL, rareSignatureTier = "T2",
      commonSignatures = NULL, rareSignatures = NULL, method = "KLD",
      exposureFilterType = "fixedThreshold", threshold_percent = 5,
      giniThresholdScaling = 10, multiStepMode = "errorReduction",
      residualNegativeProp = 0.003, minResidualMutations = NULL,
      minCosSimRareSig = 0.8, minErrorReductionPerc = 15,
      minCosSimIncrease = 0.02, useBootstrap = FALSE, nboot = 200,
      threshold_p.value = 0.05, maxRareSigsPerSample = 1, nparallel = 1,
      randomSeed = NULL, verbose = FALSE)
```

Arguments:

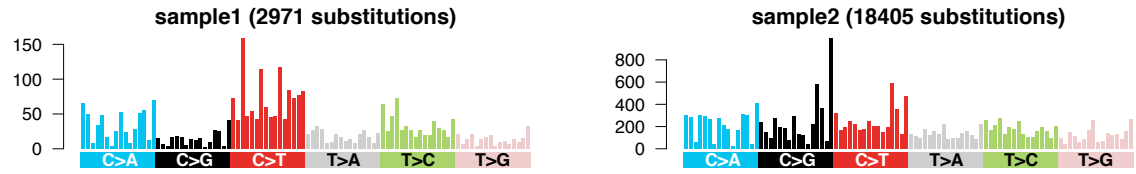
catalogues	catalogues matrix, samples as columns, channels as rows
organ	automatically sets the commonSignatures and rareSignatures parameters, which can be left as NULL. The following organs are available: "Biliary", "Bladder", "Bone_SoftTissue", "Breast", "CNS", "Colorectal", "Esophagus", "Head_neck", "Kidney", "Liver", "Lung", "Lymphoid", "Myeloid", "NET", "Oral_Oropharyngeal", "Ovary", "Pancreas", "Prostate", "Skin", "Stomach", "Uterus"
rareSignatureTier	either T1 or T2. For each organ we provide two lists of rare signatures that can be used. Tier 1 (T1) are rare signatures that were observed in the requested organ. The problem with T1 is that it may be that a signature is not observed simply because there were not enough samples for a certain organ in the particular dataset

	that was used to extract the signatures. So in general we advise to use Tier 2 (T2) signatures, which extend the rare signature to a wider number of rare signatures.
commonSignatures	signatures, signatures as columns, channels as rows. These are the signatures that are assumed to be present in most samples and will be used in the first step. Can be set automatically by specifying the organ parameter
rareSignatures	signatures, signatures as columns, channels as rows. These are the signatures that are assumed to be rarely present in a sample, at most maxRareSigsPerSample rare signatures in each sample. Can be set automatically by specifying the organ parameter and the rareSignatureTier parameter
method	KLD or NNLS
exposureFilterType	use either fixedThreshold or giniScaledThreshold. When using fixedThreshold, exposures will be removed based on a fixed percentage with respect to the total number of mutations (threshold_percent will be used). When using giniScaledThreshold each signature will used a different threshold calculated as (1-Gini(signature))*giniThresholdScaling
threshold_percent	threshold in percentage of total mutations in a sample, only exposures larger than threshold are considered
giniThresholdScaling	scaling factor for the threshold type giniScaledThreshold, which is based on the Gini score of a signature
multiStepMode	use one of the following: "constrainedFit", "partialNMF", "errorReduction", or "cossimIncrease".
residualNegativeProp	maximum proportion of mutations (w.r.t. total mutations in a sample) that can be in the negative part of a residual when using the constrained least squares fit when using multiStepMode=constrainedFit
minResidualMutations	minimum number of mutations in a residual when using constrainedFit or partialNMF. Deactivated by default.
minCosSimRareSig	minimum cosine similarity between a residual and a rare signature for considering the rare signature as a candidate for a sample when using constrainedFit or partialNMF
minErrorReductionPerc	minimum percentage of error reduction for a signature to be considered as candidate when using the errorReduction method. The error is computed as mean absolute deviation
minCosSimIncrease	minimum cosine similarity increase for a signature to be considered as candidate when using the cossimIncrease method
nboot	number of bootstraps to use, more bootstraps more accurate results
threshold_p.value	p-value to determine whether an exposure is above the threshold_percent. In other words, this is the empirical probability that the exposure is lower than the threshold
maxRareSigsPerSample	masimum number of rare signatures that should be searched in each sample. In most situations, leaving this at 1 should be enough.
nparallel	to use parallel specify >1
randomSeed	set an integer random seed
verbose	use FALSE to suppress messages

4. Examples

4.1 Using bootstrap and the Gini-based exposure filter

Consider the following two SBS mutational catalogues derived from two breast cancer whole genomes:

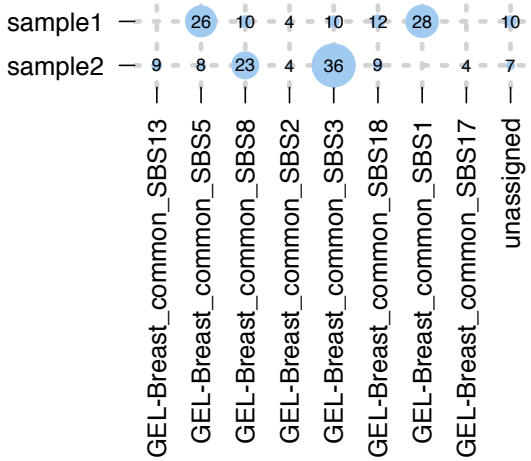


Assuming we have stored the two catalogues into a matrix C, we can apply FitMS as follows:

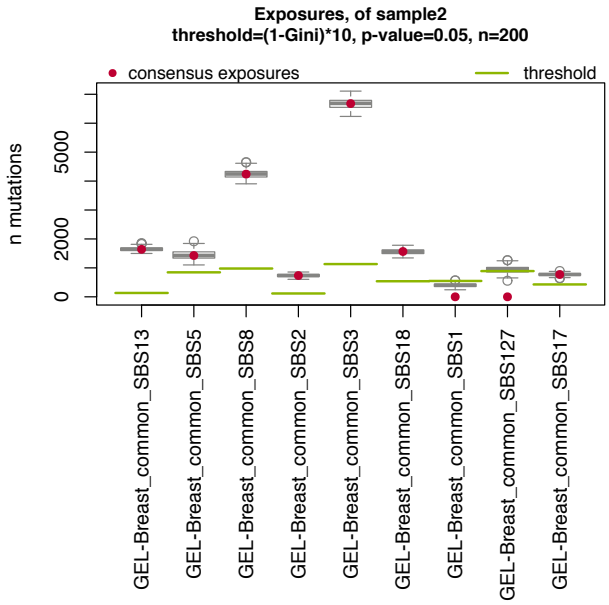
```
resObj <- FitMS(catalogues = C,
                exposureFilterType = "giniScaledThreshold",
                useBootstrap = TRUE,
                organ = "Breast",
                multiStepMode = "constrainedFit")
)
plotFitMS(resObj,outdir = "signatureFit/")
```

Specifying “Breast” as organ, automatically selected 9 GEL organ-specific common signatures and 54 rare signatures to use. Moreover, we have requested the use of bootstraps and the “constrainedFit” approach in the FitMS second step.

Among the plots obtained from executing the R script above, we can find the point estimate exposures, visualised here below as proportion of total mutations:

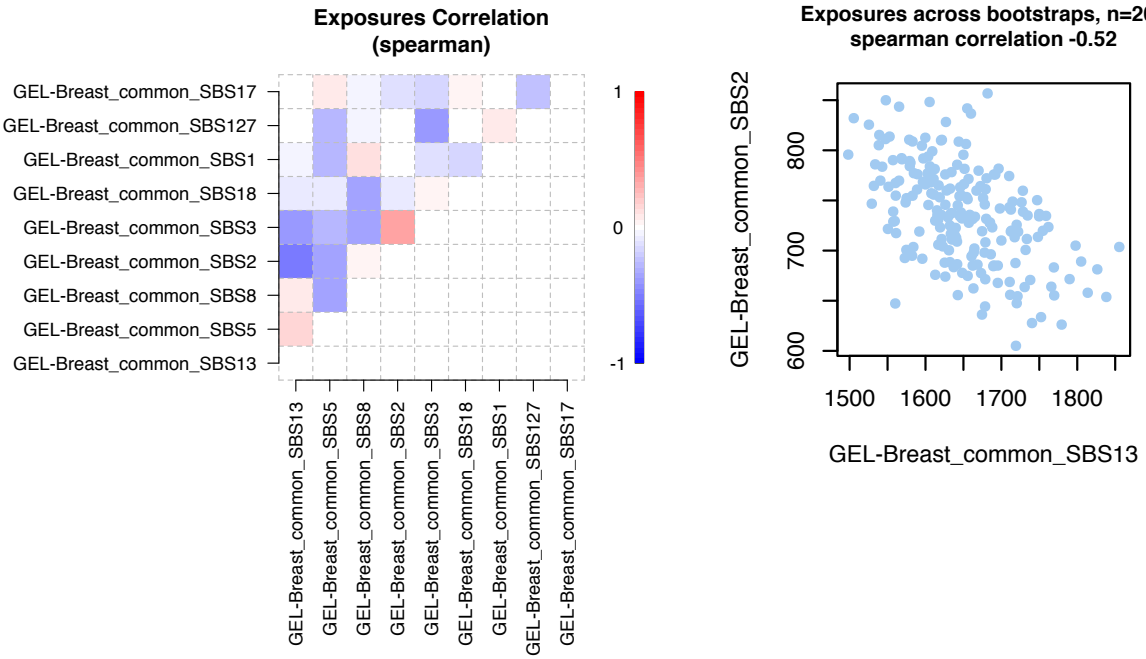


In this case, we can notice that no rare signatures were found in these samples. We can also look at the exposure distribution and how the Gini-based threshold works looking at the following plot:



In the plot above, the red dots are the point estimate exposures and the green lines indicate the thresholds computed using the Gini-based exposure filter. As expected, the threshold for SBS2 is much lower than the thresholds for SBS3 or SBS5.

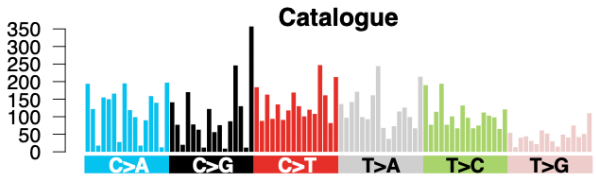
Finally, we can look at how the exposures obtained with the bootstrap method correlate:



The above plot on the left, shows the Spearman's correlation of the exposures in sample2. Some signature exposures are negatively correlated, indicating that one signature can replace the other to some extent depending on the bootstrapped catalogue. On the right, we show an example of such negative correlation, showing all the estimate exposures from 200 bootstraps.

4.2 Identify rare signatures

In the next example, we analyse a sample from PCAWG Biliary, with the following SBS mutational catalogue:



This sample is BRCA2 null, so we expect to see SBS3 and SBS8, which are usually found in samples with homologous recombination deficiency (HRD) induced by the absence of functional BRCA1 or BRCA2 proteins. In the biliary tumour type, SBS3 and SBS8 are rare signatures, so we expect FitMS to identify them.

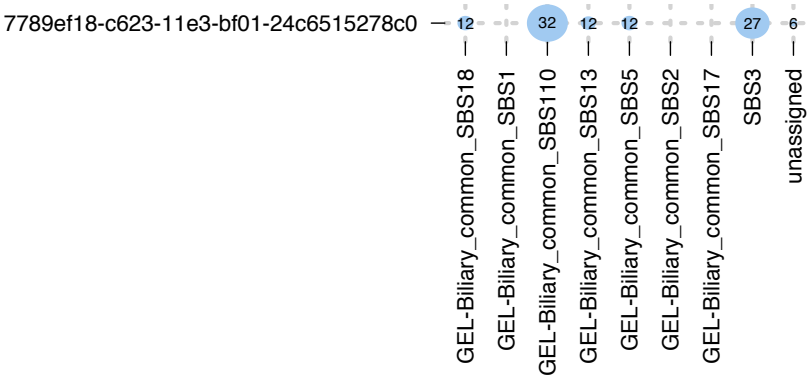
We begin by running FitMS with the following R script:

```
fitObj <- FitMS(catalogue,
                organ = "Biliary")
```

The above command will use all the default options of FitMS and is an example the minimal code one needs to write to use FitMS.

Looking at the best solution we can see that the rare signature SBS8 was identified:

However, the above solution is not the only viable solution. Looking at the other candidate solutions, we can see that SBS3 was also a good solution:



Because by default the maximum number of rare signatures in one sample is set to 1, then only one signature from either SBS3 and SBS8 is allowed. In this case, we can then increase the maximum number to two and see if both SBS3 and SBS8 together improve the solution even more. It should be noted, that because of the way FitMS is implemented, both SBS3 and SBS8 will be considered to be present only if either adding SBS8 significantly improves upon the solution with SBS3, or adding SBS3 improves upon the solution with SBS8.

We can rerun FitMS to allow two rare signatures as follows:

```
fitObj <- FitMS(catalogue,
                organ = "Biliary",
                maxRareSigsPerSample = 2)
```

As expected, FitMS reports the following best solution, which includes both SBS3 and SBS8: