

# Summary of TMB runs

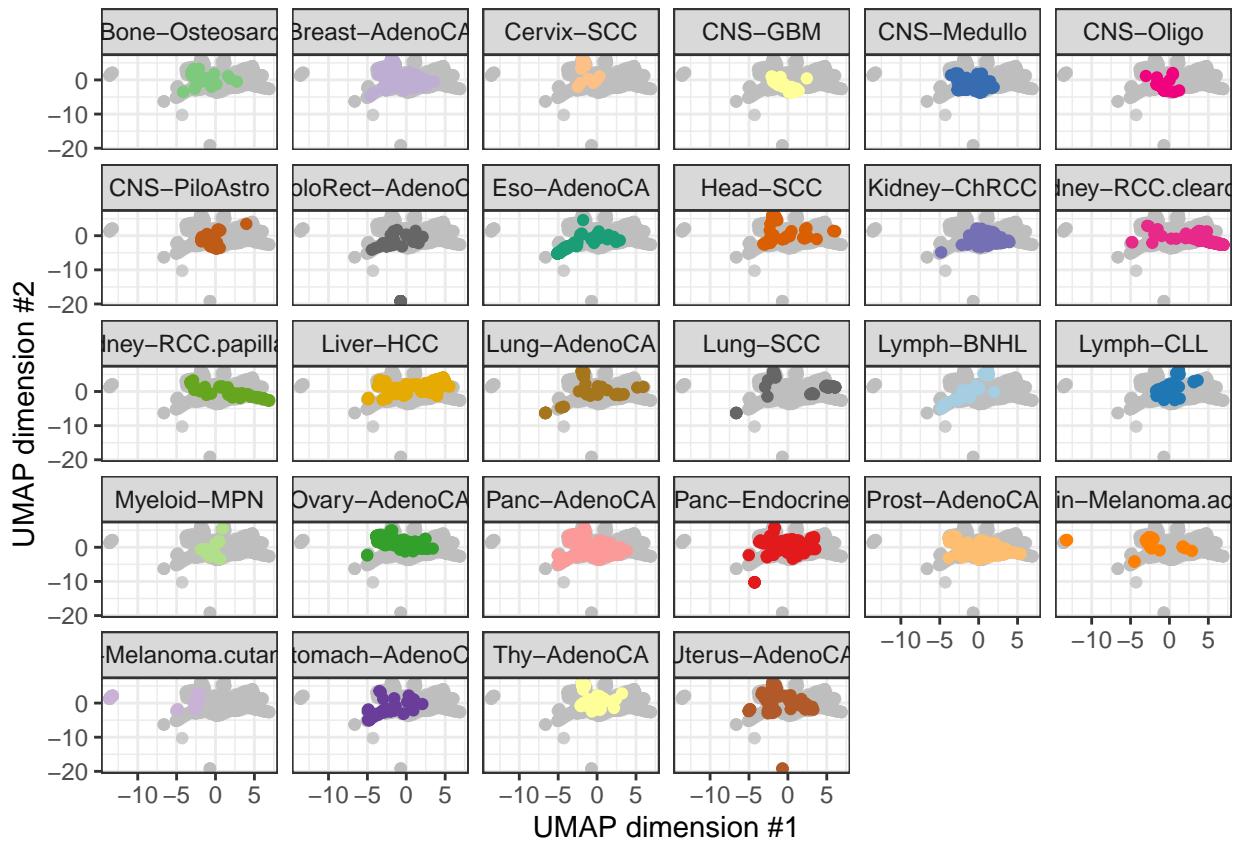
Lena Morrill

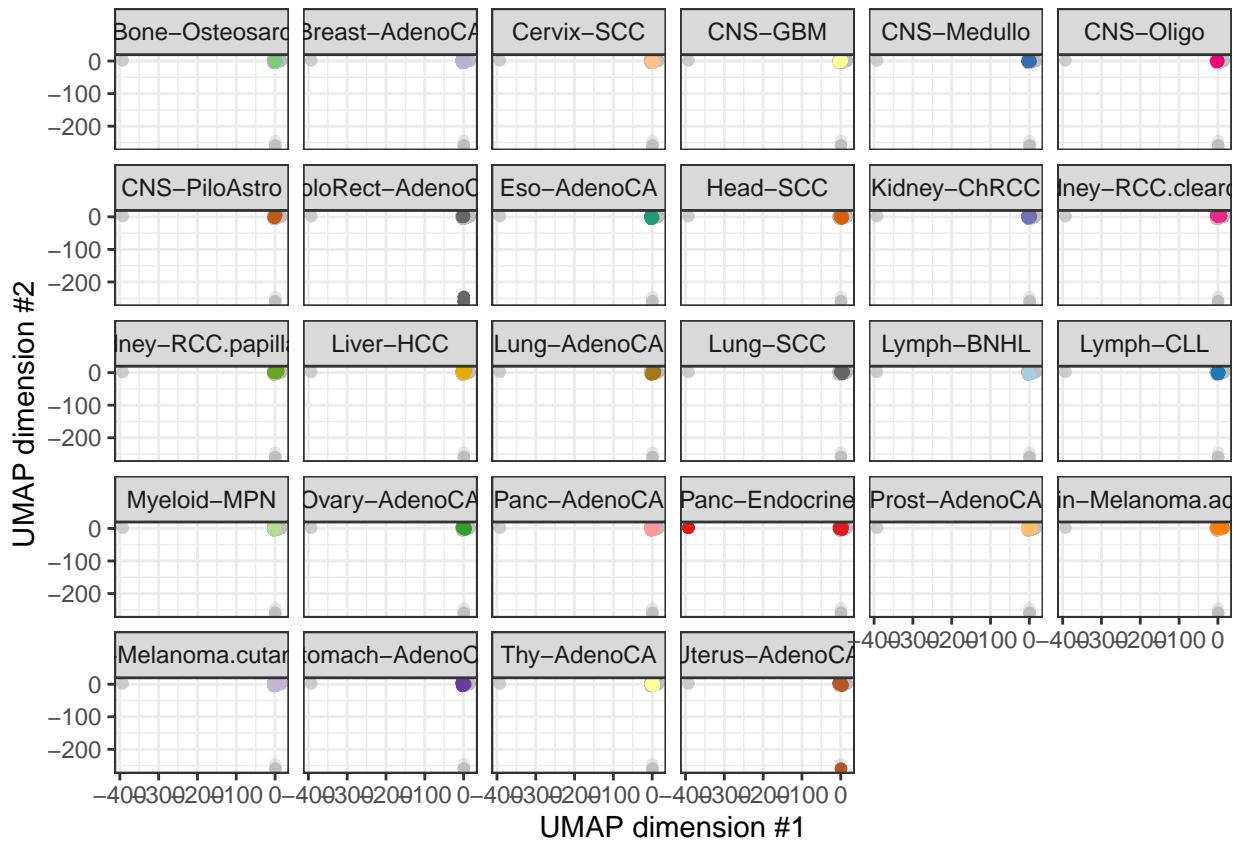
24/05/2021

```
source("../2_inference_TMB/helper_TMB.R")
source("../.../CDA_in_Cancer/code/functions/meretricious/pretty_plots/prettySignatures.R")

## Loading required package: coda
## Loading required package: MASS
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
## ## Copyright (C) 2003-2021 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
## ##
## ## Support provided by the U.S. National Science Foundation
## ## (Grants SES-0350646 and SES-0350613)
## ##

## Error in slot(i, "count_matrices_all") :
##   cannot get a slot ("count_matrices_all") from an object of type "logical"
## Error in slot(i, "count_matrices_all") :
##   cannot get a slot ("count_matrices_all") from an object of type "logical"
```





## Contents

<b>Information about models</b>	<b>6</b>
Default order of categories for each model . . . . .	6
<b>General results of all models</b>	<b>7</b>
P-values for all cancer types . . . . .	8
All betas with SBS1 as baseline . . . . .	9
<b>Analysis per cancer type</b>	<b>12</b>
Bone osteosarcoma . . . . .	12
Barplot and general statistics . . . . .	12
Convergence table . . . . .	12
Re-running of fitting . . . . .	13
Potentially problematic signatures . . . . .	13
Betas . . . . .	13
Covariance matrices . . . . .	15
Simulation under inferred data . . . . .	16
Ranked plot for coverage . . . . .	16
Signatures from mutSigExtractor . . . . .	17
Breast-AdenoCA . . . . .	19
Barplot and general statistics . . . . .	19
Convergence table . . . . .	20
Re-running of fitting . . . . .	21

Potentially problematic signatures	21
Betas	22
Covariance matrices	25
Simulation under inferred data	25
Ranked plot for coverage	26
Signatures from mutSigExtractor	27
Cervix-SCC	28
Barplot and general statistics	28
Convergence table	29
Potentially problematic signatures	30
Betas	30
Covariance matrices	31
Simulation under inferred data	32
Ranked plot for coverage	33
Signatures from mutSigExtractor	34
CNS-GBM	35
Barplot and general statistics	35
Convergence table	36
Re-running of fitting	37
Potentially problematic signatures	37
Betas	37
Covariance matrices	40
Simulation under inferred data	40
Ranked plot for coverage	41
Signatures from mutSigExtractor	42
CNS-Medullo	44
Barplot and general statistics	44
Convergence table	45
Potentially problematic signatures	46
Betas	46
Covariance matrices	49
Simulation under inferred data	49
Ranked plot for coverage	50
Signatures from mutSigExtractor	51
CNS-Oligo	53
Barplot and general statistics	53
Convergence table	54
CNS-PiloAstro	55
ColoRect-AdenoCA	55
Eso-AdenoCA	55
Head-SCC	55
Kidney-ChRCC	55
Kidney-RCC.clearcell	55
Kidney-RCC.papillary	55
Liver-HCC	55
Lung-AdenoCA	55
Lung-SCC	55
Lymph-BNHL	55
Lymph-CLL	55
Myeloid-MPN	55
Ovary-AdenoCA	55

Panc-AdenoCA . . . . .	55
Panc-Endocrine . . . . .	55
Prost-AdenoCA . . . . .	55
Skin-Melanoma.acral . . . . .	55
Skin-Melanoma.cutaneous . . . . .	55
Stomach-AdenoCA . . . . .	55
Thy-AdenoCA . . . . .	55
Uterus-AdenoCA . . . . .	55
<b>All p-values for non-exogenous signatures</b>	<b>55</b>
<b>Dirichlet-Multinomial Mixtures</b>	<b>56</b>
Comparison of signature exposures with QP and mutsigextractor . . . . .	62

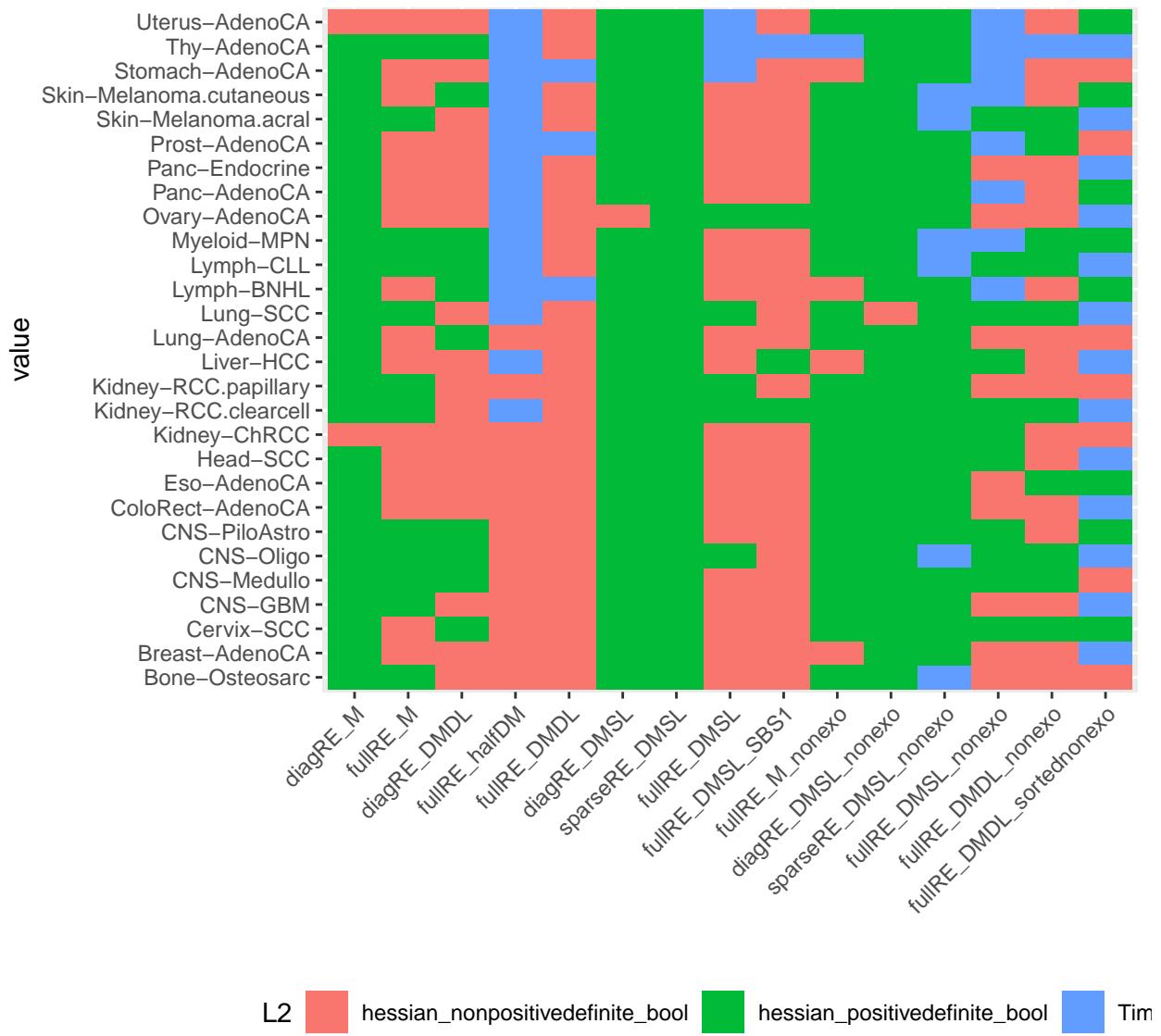
## Information about models

### Default order of categories for each model

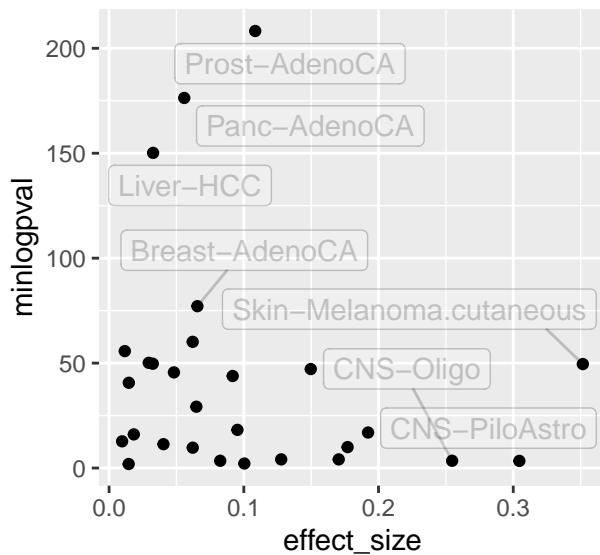
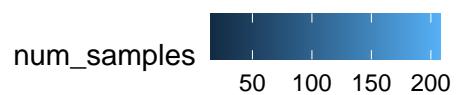
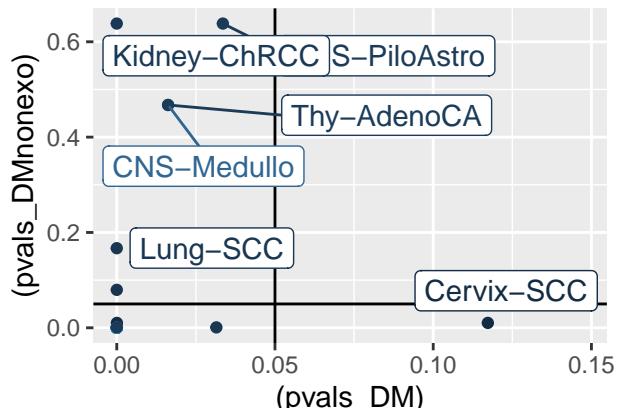
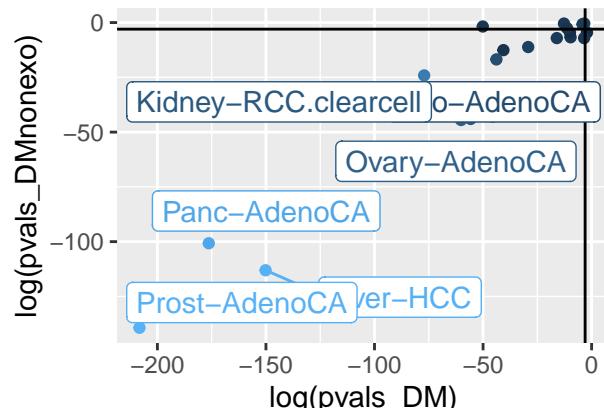
Name model	Extension	Sorted	File in which they were created
fullREDMsinglelambda	fullRE_DMSL_	Not sorted	run_TMB_PCAWG.R
fullREDMsinglelambda2	fullRE_DMSL2_	Sorted	run_TMB_PCAWG.R
diagREDMsinglelambda	diagRE_DMSL_	Unknown	run_TMB_PCAWG.R
		Sorted in previous version of	
fullRE_M	fullRE_M_		run_TMB_PCAWG.R wrapper_run_TMB
		Sorted in previous version of	
diagRE_DM	diagRE_DM_		run_TMB_PCAWG.R wrapper_run_TMB
		Sorted in previous version of	
fullRE_DM	fullRE_DM_		run_TMB_PCAWG.R wrapper_run_TMB
		Sorted	
sparseRE_DMSL2	sparseRE_nonexo_DMSL_	Sorted	find_subset_signatures.R
fullREDMsinglelambda	fullRE_nonexo_DMSL_	Not sorted	find_subset_signatures.R
fullRE_M	fullRE_nonexo_M_	Not sorted	find_subset_signatures.R
diagREDMsinglelambda	diagRE_nonexo_DMSL_	Not sorted	find_subset_signatures.R
fullRE_DM	fullRE_nonexo_DM_	Not sorted	find_subset_signatures.R
diagREDMsinglelambda	diagRE_DMSL_	Not sorted	find_subset_signatures.R

## General results of all models

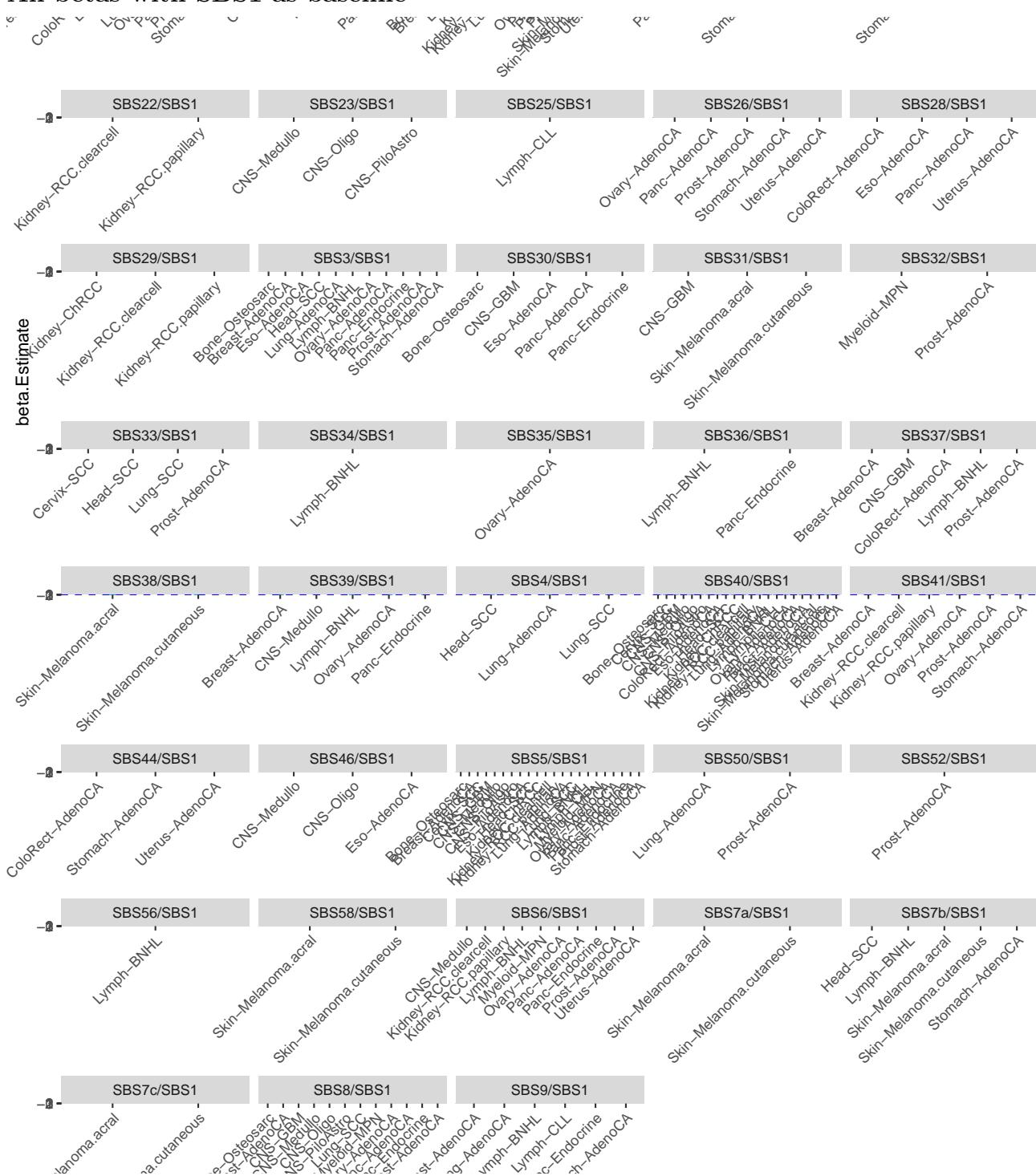
Check the results of all of the models

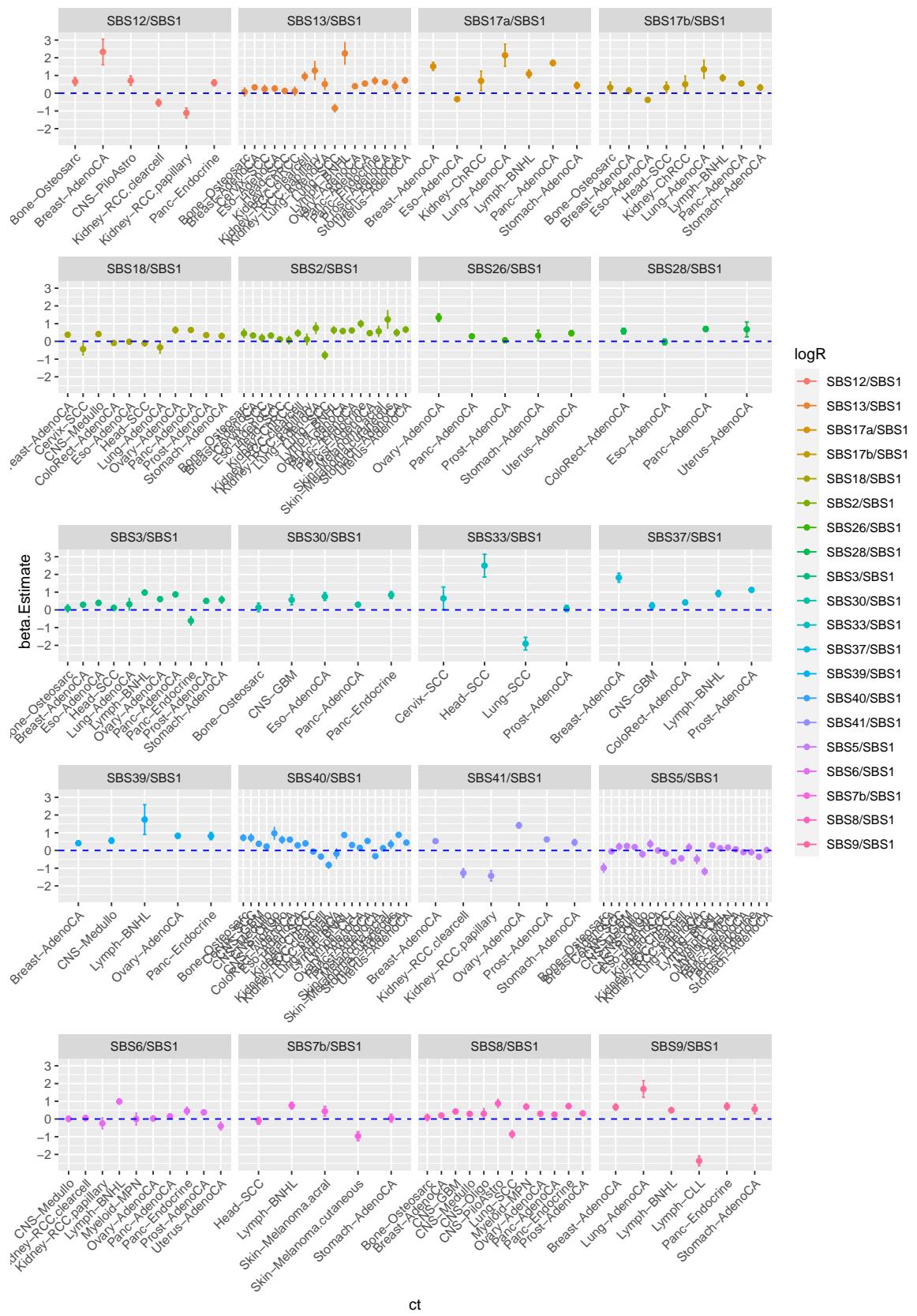


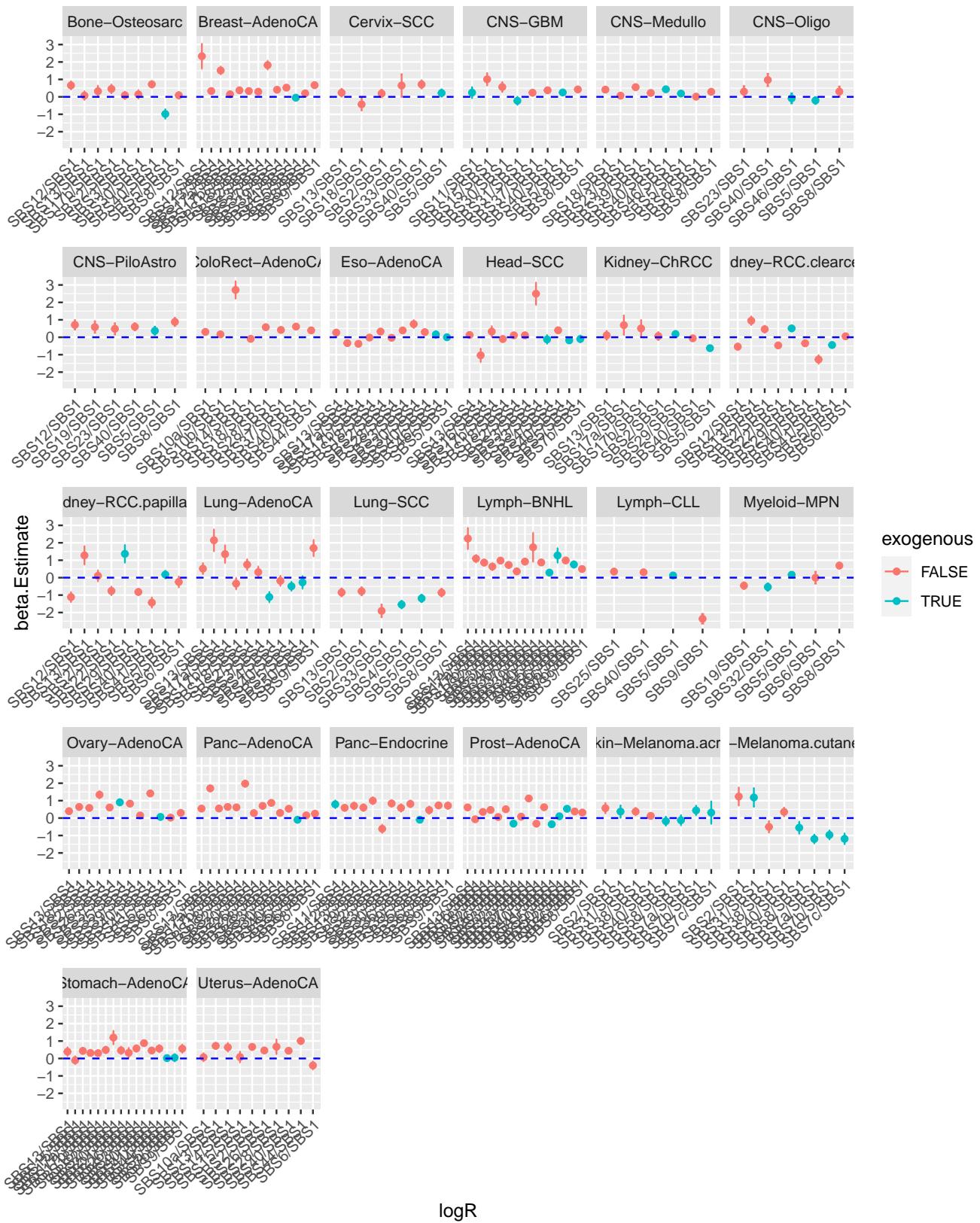
## P-values for all cancer types



### All betas with SBS1 as baseline







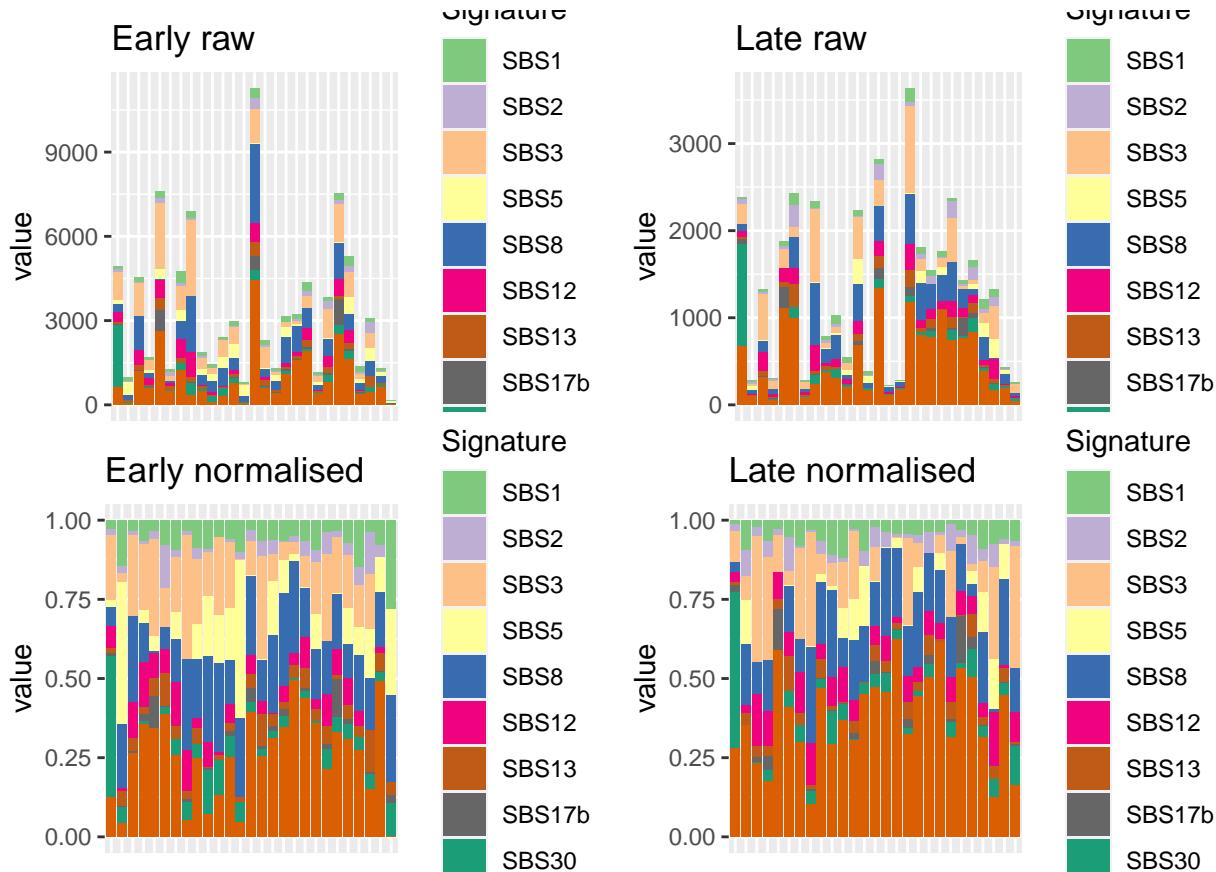
## Analysis per cancer type

### Bone osteosarcoma

#### Barplot and general statistics

```
## [1] 27
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
```



The number of samples and signatures is:

```
## [1] 54 10
```

The signatures are:

```
## [1] "SBS1"   "SBS2"   "SBS3"   "SBS5"   "SBS8"   "SBS12"  "SBS13"  "SBS17b"
## [9] "SBS30"  "SBS40"
```

#### Convergence table

We only have converged results for the multinomial with full RE, and the DM with a single lambda (diag and full RE). It is the same for nonexogenous signatures.

##	value	L2	L1
----	-------	----	----

```

## 1 Bone-Osteosarc    hessian_positivedefinite_bool          diagRE_M
## 2 Bone-Osteosarc    hessian_positivedefinite_bool          fullRE_M
## 3 Bone-Osteosarc  hessian_nonpositivedefinite_bool        diagRE_DMDL
## 4 Bone-Osteosarc  hessian_nonpositivedefinite_bool        fullRE_halfDM
## 5 Bone-Osteosarc  hessian_nonpositivedefinite_bool        fullRE_DMDL
## 6 Bone-Osteosarc    hessian_positivedefinite_bool        diagRE_DMSL
## 7 Bone-Osteosarc    hessian_positivedefinite_bool        sparseRE_DMSL
## 8 Bone-Osteosarc  hessian_nonpositivedefinite_bool        fullRE_DMSL
## 9 Bone-Osteosarc  hessian_nonpositivedefinite_bool        fullRE_DMSL_SBS1
## 10 Bone-Osteosarc   hessian_positivedefinite_bool        fullRE_M_nonexo
## 11 Bone-Osteosarc   hessian_positivedefinite_bool        diagRE_DMSL_nonexo
## 12 Bone-Osteosarc                      Timeout        sparseRE_DMSL_nonexo
## 13 Bone-Osteosarc  hessian_nonpositivedefinite_bool        fullRE_DMSL_nonexo
## 14 Bone-Osteosarc  hessian_nonpositivedefinite_bool        fullRE_DMDL_nonexo
## 15 Bone-Osteosarc  hessian_nonpositivedefinite_bool        fullRE_DMDL_sortednonexo

```

### Re-running of fitting

```
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```

If we use the values of the fullRE M as initial values for the fullRE DM, we also don't get convergence:

```
## [1] FALSE
```

### Potentially problematic signatures

We notice that we have several signatures with low exposures, and many zero exposures

```
colSums(obj_Bone_Osteosarc$Y == 0)/nrow(obj_Bone_Osteosarc$Y)
```

```

##      SBS1      SBS2      SBS3      SBS5      SBS8      SBS12     SBS13
## 0.00000000 0.03703704 0.14814815 0.37037037 0.01851852 0.09259259 0.00000000
##      SBS17b     SBS30     SBS40
## 0.37037037 0.12962963 0.01851852

```

```
colSums(obj_Bone_Osteosarc$Y)/sum(obj_Bone_Osteosarc$Y)
```

```

##      SBS1      SBS2      SBS3      SBS5      SBS8      SBS12     SBS13
## 0.05099661 0.03376971 0.17876022 0.05053018 0.17164713 0.07538325 0.04159022
##      SBS17b     SBS30     SBS40
## 0.02866227 0.06128922 0.30737119

```

E.g.

- SBS17b is 0 in 37% of cases and has an overall exposure of 2.9%
- SBS30 is 0 in 13% of cases and overall has an exposure of only 6.1%
- SBS5 is 0 in 37% of cases and has an overall exposure of 5.1%

### Betas

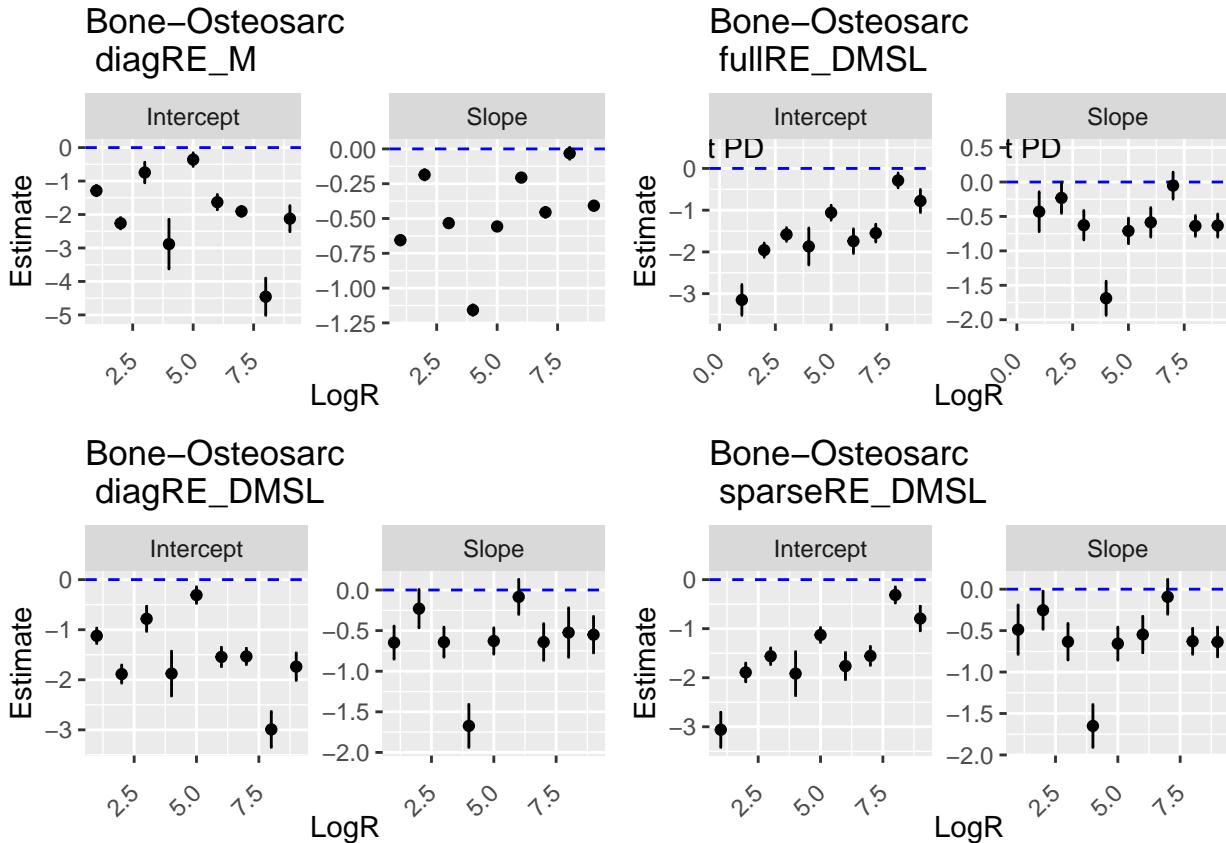
```

ct <- "Bone-Osteosarc"

grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL'))),

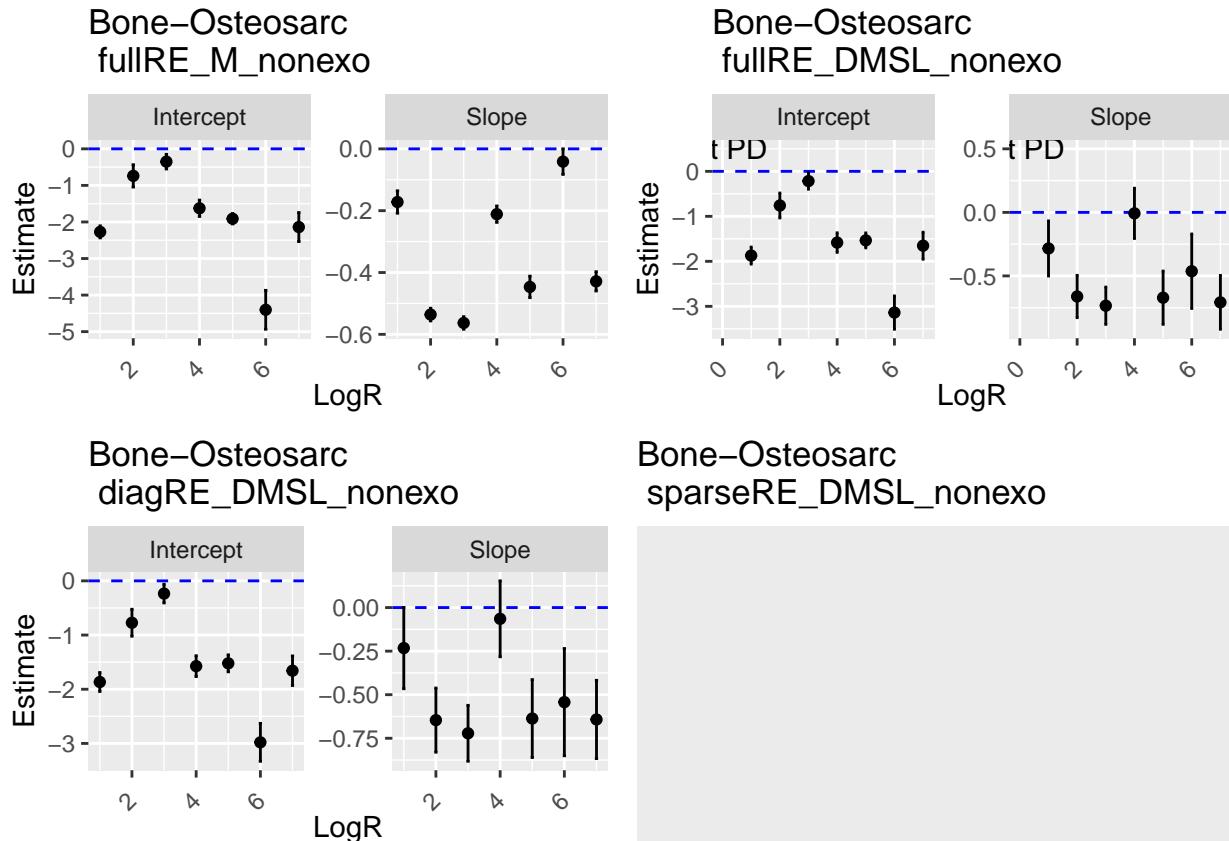
```

```
plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),  
plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)
```



```
grid.arrange(  
  plot_betas(fullRE_M_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_M_nonexo')),  
  plot_betas(fullRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL_nonexo')),  
  plot_betas(diagRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL_nonexo')),  
  plot_betas(sparseRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL_nonexo')), nrow=2)
```

```
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```



```

## Warning in select_slope_2(which(names(i$par.fixed) == "beta"), verbatim
## = verbatim): As per 27 August it seems clear that this version, and not
## <select_slope>, is correct

## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma**(1/2) has now been replaced by (as we
## had before sometime in November) sigma

```

We use the results from the diagonal single lambda DM to test for differential abundance, giving a p-value of  $3.8923434 \times 10^{-5}$ .

### Covariance matrices

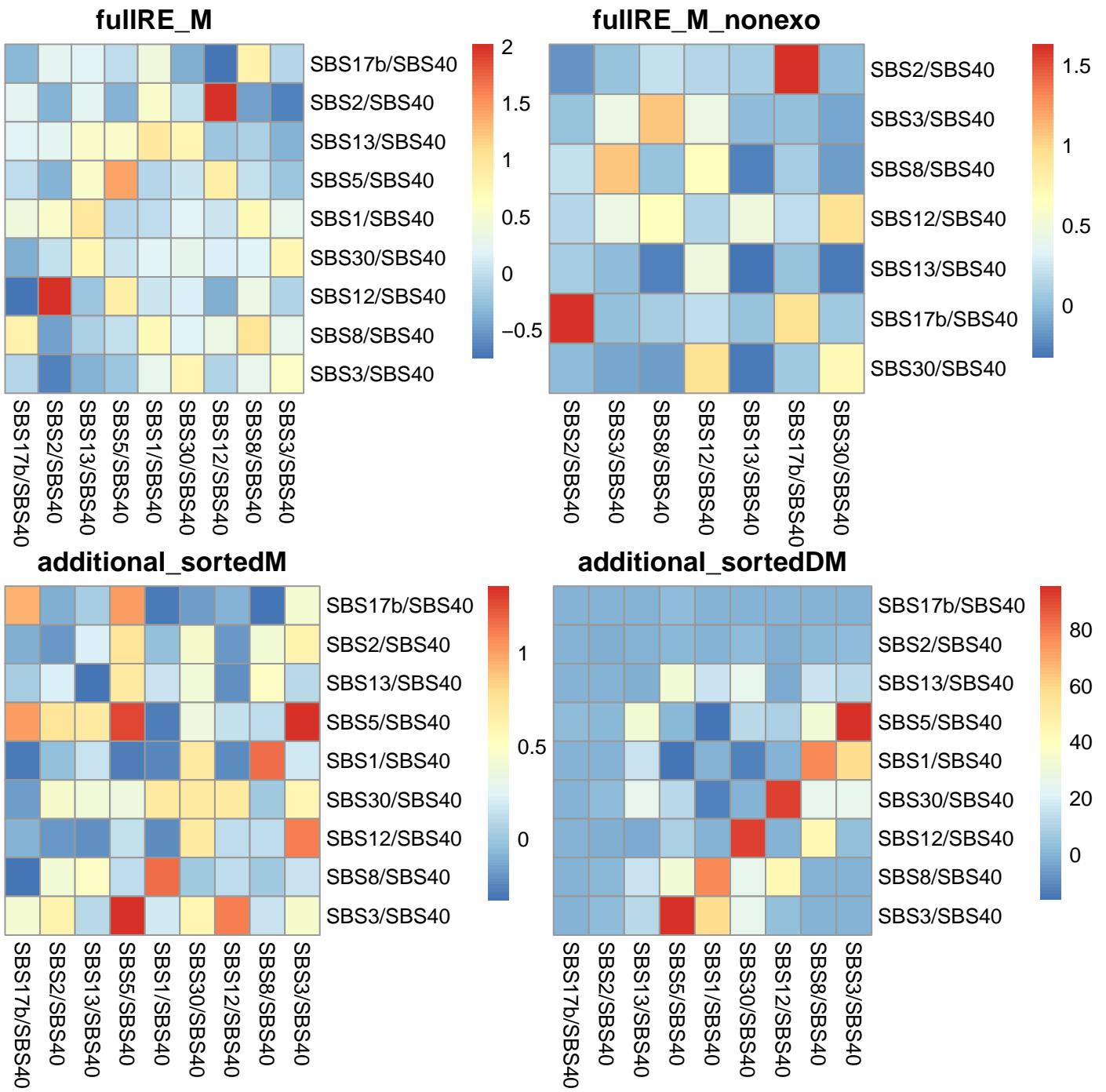
```

ct <- "Bone-Osteosarc"
additional_sortedM <- list()
additional_sortedDM <- list()
additional_sortedM[[ct]] <- sortedM
additional_sortedDM[[ct]] <- sortedDM

```

Note that sortedDM did not converge.

Nevertheless, both versions of fullRE M – both of which converged and use the same baseline – give very different covariances matrices.



## Simulation under inferred data

Have not been able to simulate

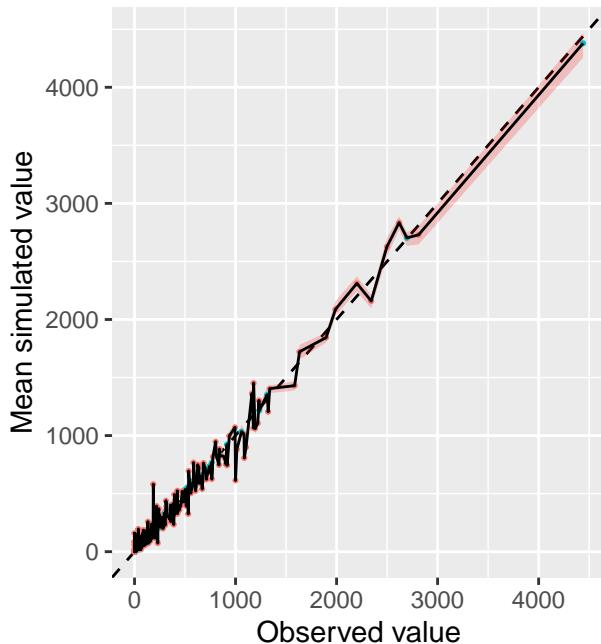
## Ranked plot for coverage

```

ct <- "Bone-Osteosarc"
integer_overdispersion_param_DMSL <- 1
obj_Bone_Osteosarc_nonexo <- give_subset_sigs_TMBobj(obj_Bone_Osteosarc, sigs_to_remove = nonexogenous$V1)
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL,
data_object = obj_Bone_Osteosarc_nonexo,
print_plot = F, nreps = 20, model = "M")), function(i){
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )})[[1]],
data_object = obj_Bone_Osteosarc_nonexo,
loglog = F, title = 'obj_Bone_Osteosarc (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_Bone_Osteosarc_nonexo,
print_plot = F, nreps = 20, model = "DMSL", integer_overdispersion_param = integer_overdispersion_param_DMSL)),
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )})[[1]],
data_object = obj_Bone_Osteosarc_nonexo,
loglog = F, title = 'obj_Bone_Osteosarc (DMSL)', ncol=2)

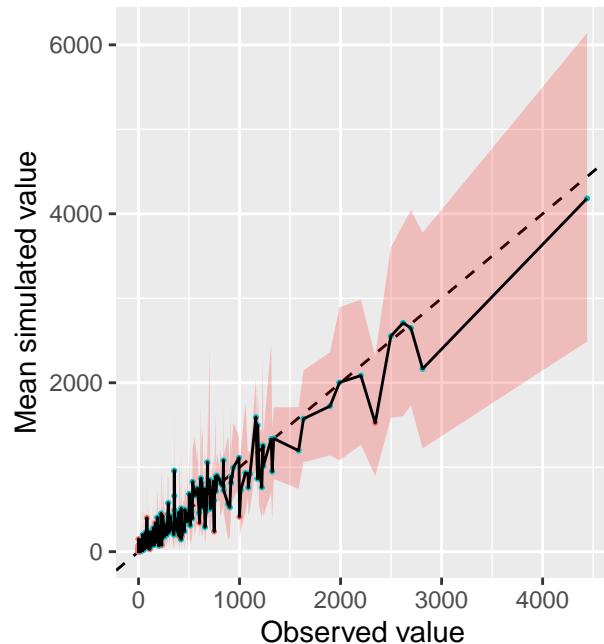
```

**obj\_Bone\_Osteosarc (M)**  
FALSE:268; TRUE:164



col    • FALSE    • TRUE

**obj\_Bone\_Osteosarc (DMSL)**  
FALSE:73; TRUE:359



col    • FALSE    • TRUE

73/359=20% of values are not included in the confidence interval of the DMSL.

### Signatures from mutSigExtractor

The signatures from mutSigExtractor are a bit more chaotic:

```

obj_Bone_Osteosarc_mutSigExtractor <- load_PCAWG(ct = ct, typedata = "signaturesmutSigExtractor",
path_to_data = "../data/")

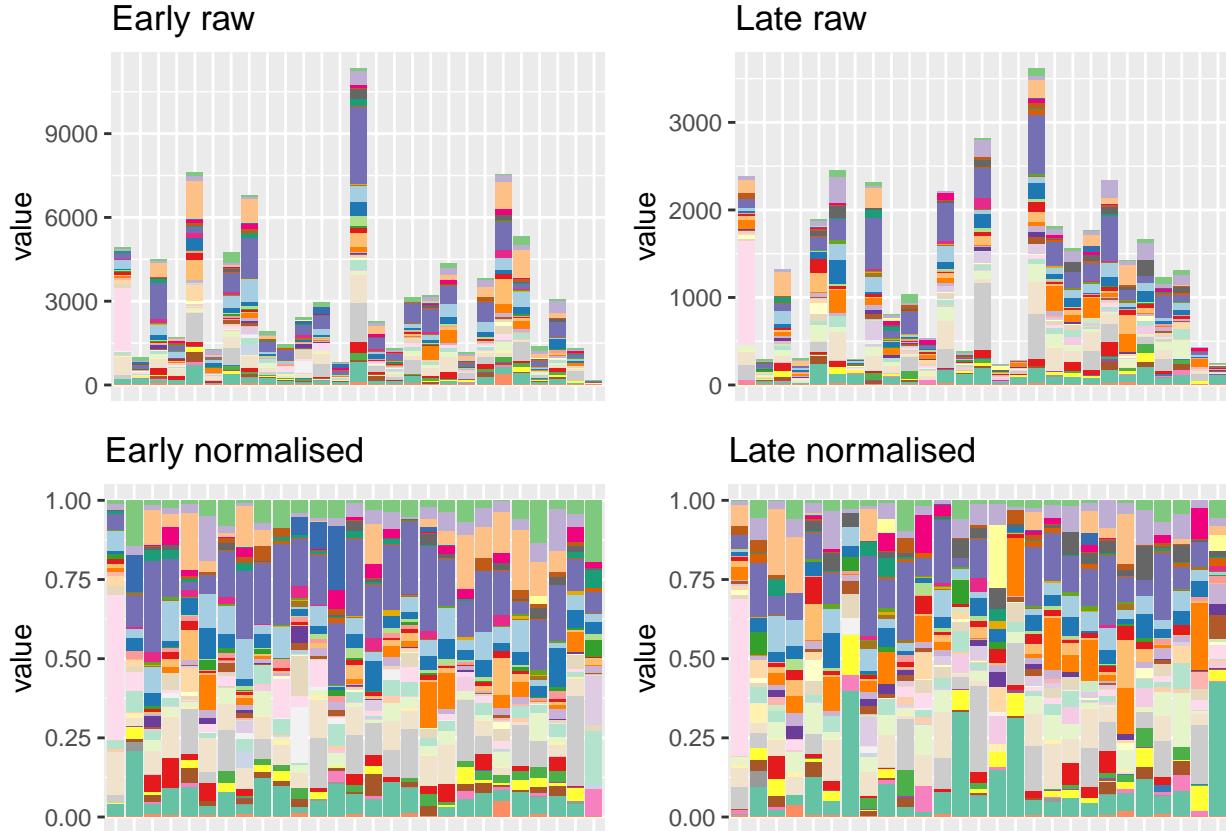
```

```

## [1] 27
give_barplot_from_obj(obj = obj_Bone_Osteosarc_mutSigExtractor, legend_on = FALSE)

## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27
## Creating plot... it might take some time if the data are large. Number of samples: 27

```



Exposures sorted by increasing number of mutations

```

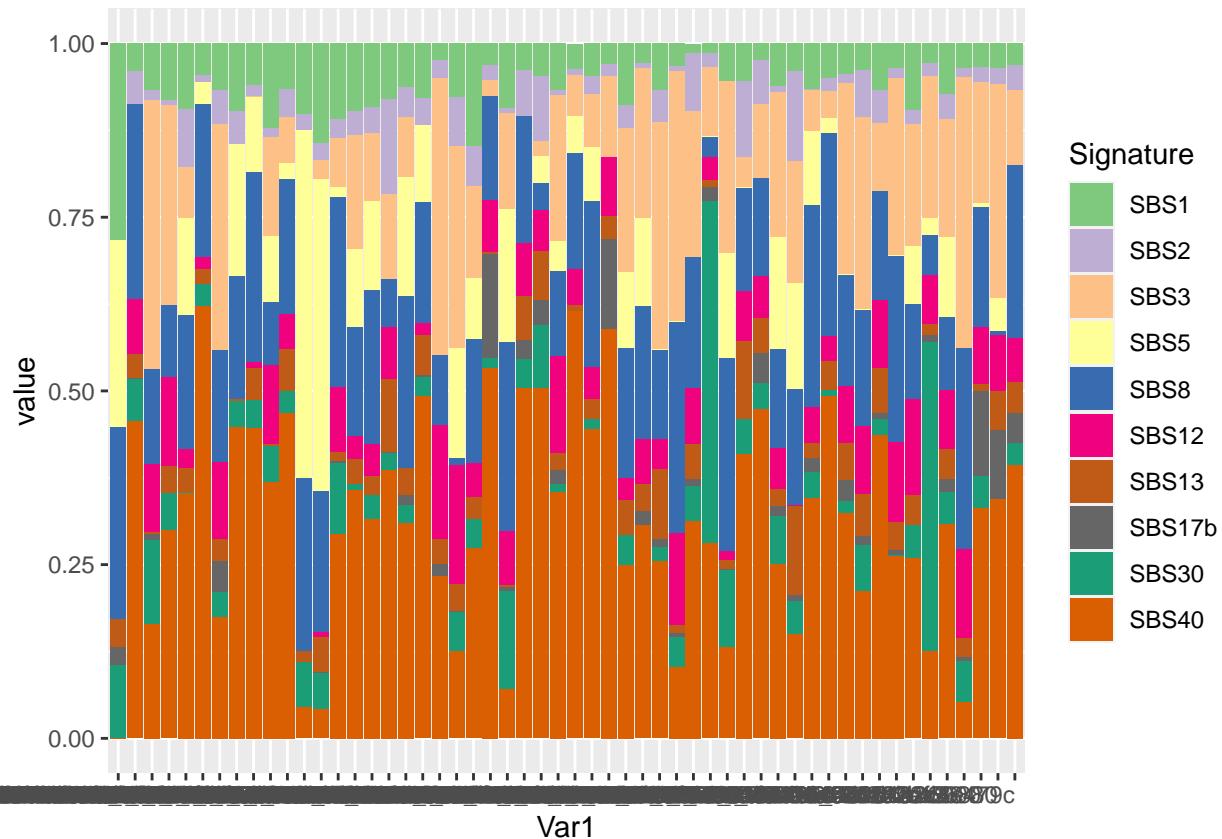
createBarplot(normalise_rw(non_duplicated_rows(obj_Bone_Osteosarc$Y)),
              order_labels = names(sort(rowSums(non_duplicated_rows(obj_Bone_Osteosarc$Y)),
                                         decreasing = F)))

```

```

## Creating plot... it might take some time if the data are large. Number of samples: 54

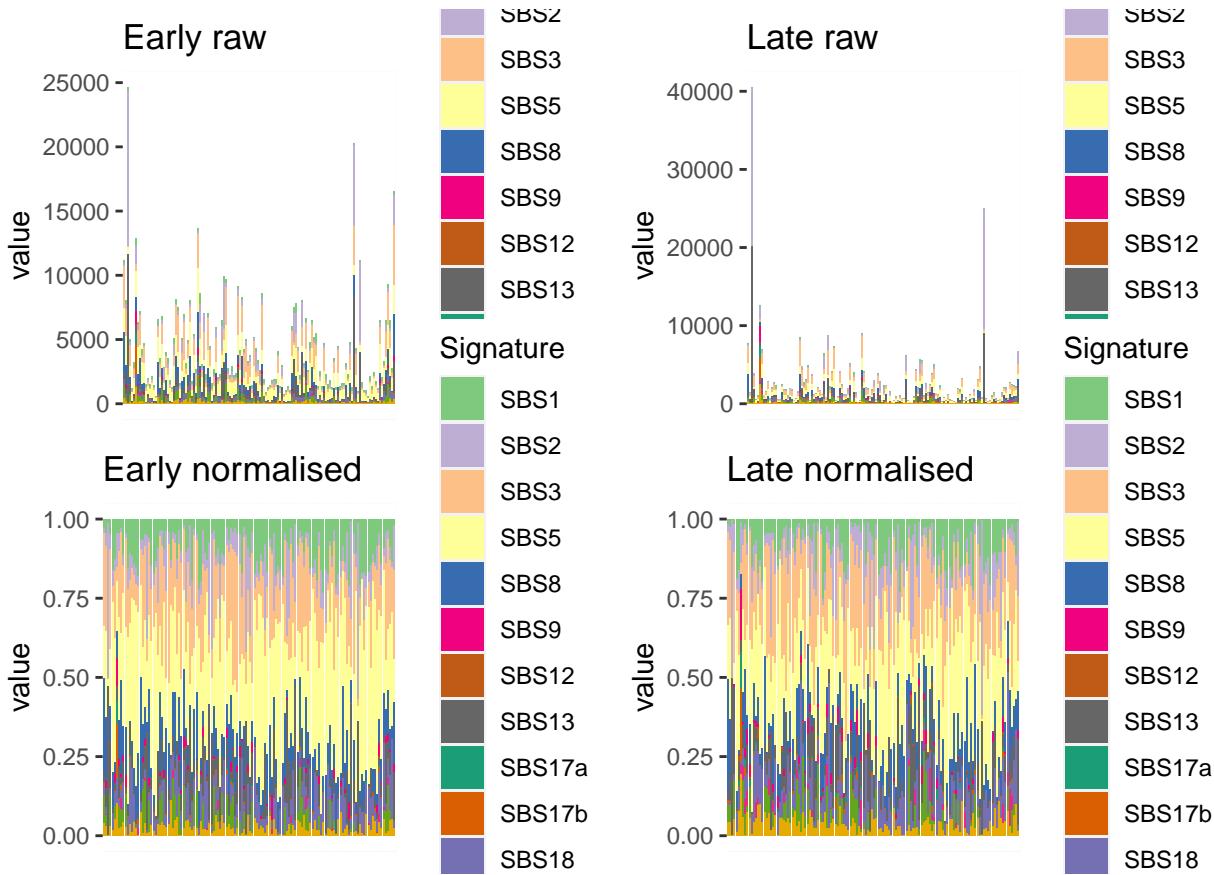
```



## Breast-AdenoCA

### Barplot and general statistics

```
## [1] 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
```



There are many signatures, and also many samples.

The number of samples and signatures is:

```
## [1] 272 14
```

The signatures are:

```
## [1] "SBS1"   "SBS2"   "SBS3"   "SBS5"   "SBS8"   "SBS9"   "SBS12"  "SBS13"
## [9] "SBS17a" "SBS17b" "SBS18"  "SBS37"  "SBS39"  "SBS41"
```

### Convergence table

We only have converged results for the diagRE\_DMSL, with diagonal or sparse covariance structure, and diagonal M. This is probably due to the very high number of signatures, which make it impossible to infer the whole covariance structure.

##	value	L2	L1
## 1	Breast-AdenoCA   hessian_positivedefinite_bool		diagRE_M
## 2	Breast-AdenoCA hessian_nonpositivedefinite_bool		fullRE_M
## 3	Breast-AdenoCA hessian_nonpositivedefinite_bool		diagRE_DMDL
## 4	Breast-AdenoCA hessian_nonpositivedefinite_bool		fullRE_halfDM
## 5	Breast-AdenoCA hessian_nonpositivedefinite_bool		fullRE_DMDL
## 6	Breast-AdenoCA   hessian_positivedefinite_bool		diagRE_DMSL
## 7	Breast-AdenoCA   hessian_positivedefinite_bool		sparseRE_DMSL
## 8	Breast-AdenoCA hessian_nonpositivedefinite_bool		fullRE_DMSL
## 9	Breast-AdenoCA hessian_nonpositivedefinite_bool		fullRE_DMSL_SBS1

```

## 10 Breast-AdenoCA hessian_nonpositivedefinite_bool      fullRE_M_nonexo
## 11 Breast-AdenoCA    hessian_positivedefinite_bool     diagRE_DMSL_nonexo
## 12 Breast-AdenoCA    hessian_positivedefinite_bool     sparseRE_DMSL_nonexo
## 13 Breast-AdenoCA hessian_nonpositivedefinite_bool      fullRE_DMSL_nonexo
## 14 Breast-AdenoCA hessian_nonpositivedefinite_bool      fullRE_DMDL_nonexo
## 15 Breast-AdenoCA                           Timeout fullRE_DMDL_sortednonexo

```

### Re-running of fitting

If we use the values of the diagRE M as initial values for the diagRE DM, we see that it has converged. This is probably due to a combination of things: we are using the optimiser nlmminb (better in general than the alternative, optim) and we are starting with these - better - values, and we are sorting the columns so that the category with highest total value is the baseline.

```

## [1] TRUE
ct <- "Breast-AdenoCA"
additional_sorteddiagM <- list()
additional_sorteddiagDM <- list()
additional_sorteddiagM[[ct]] <- sortedM_Breast_Adeno
additional_sorteddiagDM[[ct]] <- sortedDM_Breast_Adeno

```

### Potentially problematic signatures

We notice that we have several signatures with low exposures, and many zero exposures

```

colSums(obj_Breast_AdenoCA$Y == 0)/nrow(obj_Breast_AdenoCA$Y)

##          SBS1        SBS2        SBS3        SBS5        SBS8        SBS9
## 0.000000000 0.000000000 0.025735294 0.007352941 0.088235294 0.562500000
##          SBS12       SBS13      SBS17a      SBS17b      SBS18      SBS37
## 0.955882353 0.073529412 0.709558824 0.500000000 0.036764706 0.772058824
##          SBS39       SBS41
## 0.599264706 0.084558824

colSums(obj_Breast_AdenoCA$Y)/sum(obj_Breast_AdenoCA$Y)

##          SBS1        SBS2        SBS3        SBS5        SBS8        SBS9
## 0.0553410311 0.1376261991 0.1993274971 0.2185906789 0.0969490005 0.0132833987
##          SBS12       SBS13      SBS17a      SBS17b      SBS18      SBS37
## 0.0003532317 0.1360853961 0.0036266519 0.0081714966 0.0531199688 0.0057240307
##          SBS39       SBS41
## 0.0402034279 0.0315979909

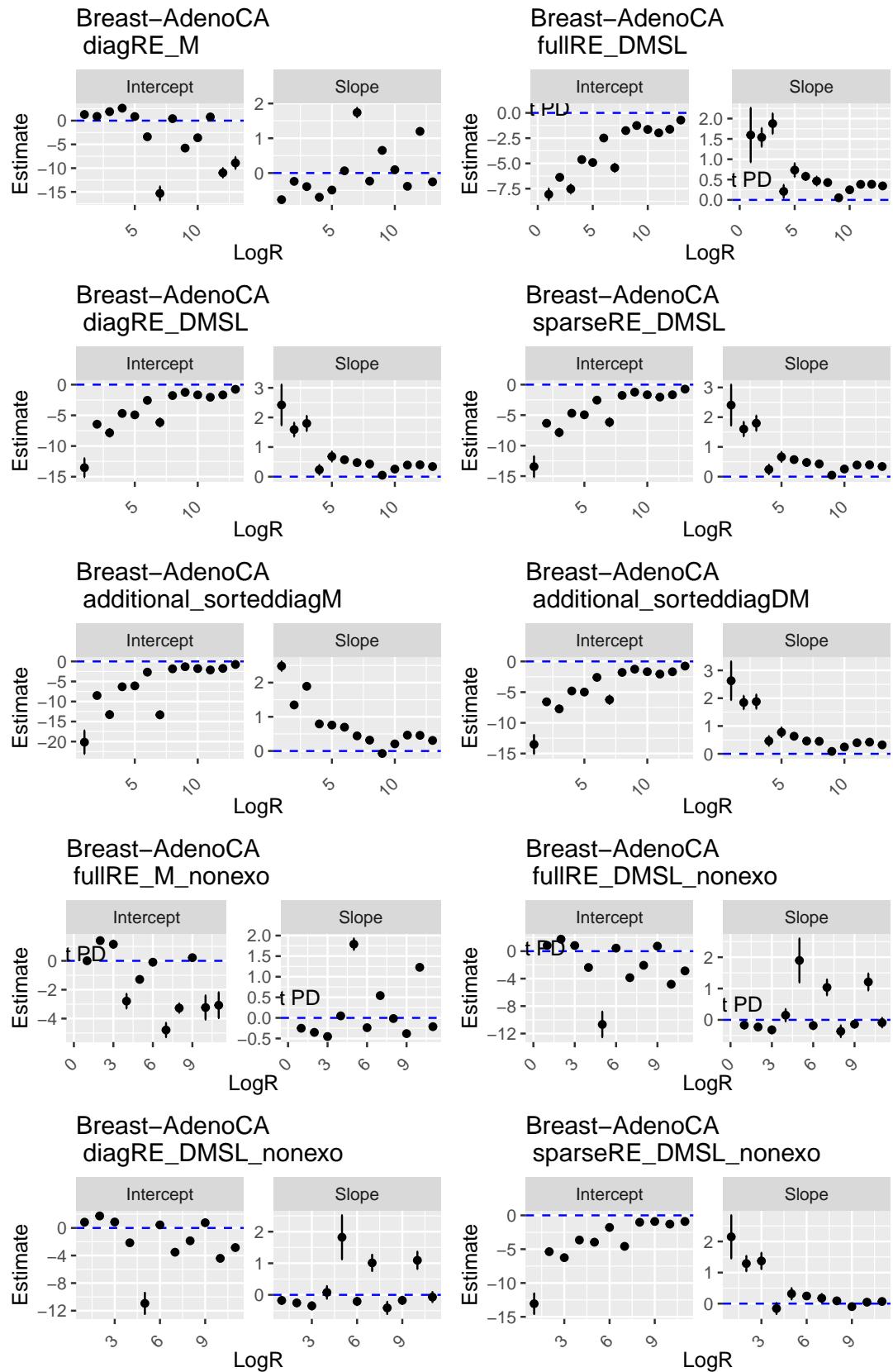
```

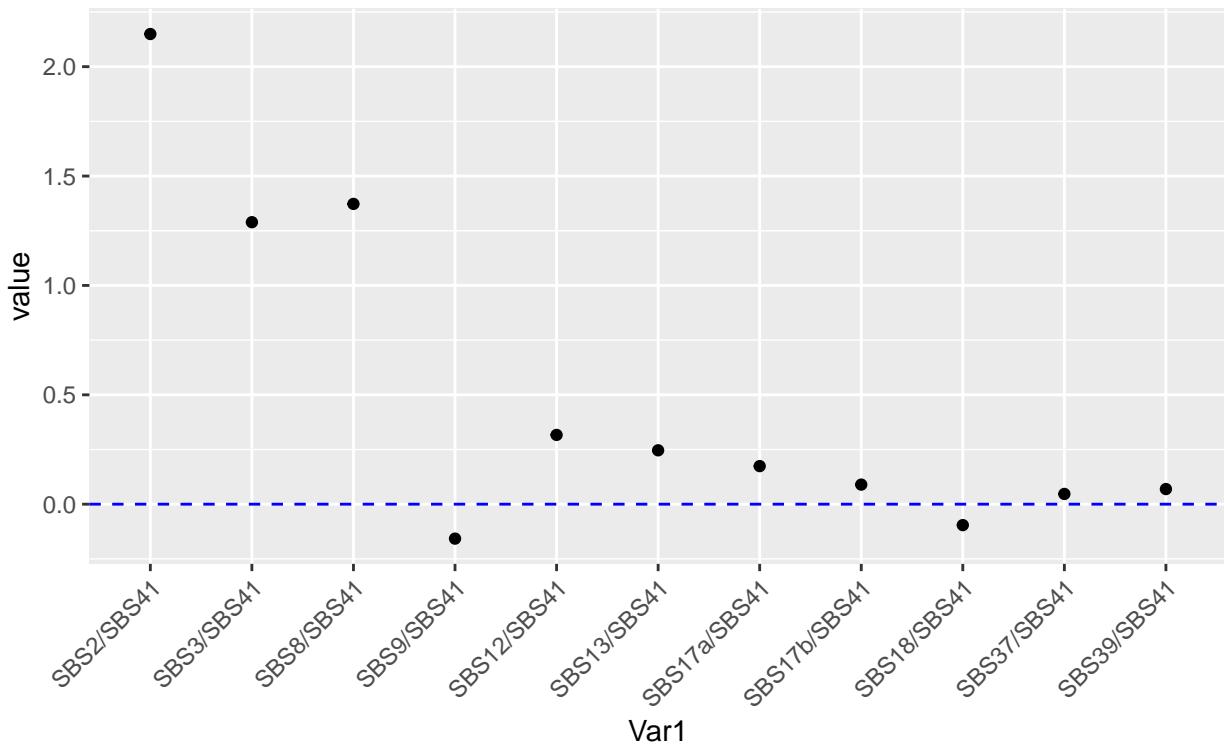
E.g.

- SBS9 is 0 in 56.2% of cases and has an overall exposure of 1.3%
- SBS12 is 0 in 95.6% of cases and has an overall exposure of 0%
- SBS17a is 0 in 71% of cases and has an overall exposure of 0.4%
- SBS17b is 0 in 50% of cases and has an overall exposure of 0.8%
- SBS37 is 0 in 77.2% of cases and has an overall exposure of 0.6%
- SBS39 is 0 in 59.9% of cases and has an overall exposure of 4%

## Betas

```
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
## Warning in sqrt(as.numeric(object$diag.cov.random)): NaNs produced
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
## Warning in sqrt(as.numeric(object$diag.cov.random)): NaNs produced
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```





```

## Warning in select_slope_2(which(names(i$par.fixed) == "beta"), verbatim
## = verbatim): As per 27 August it seems clear that this version, and not
## <select_slope>, is correct

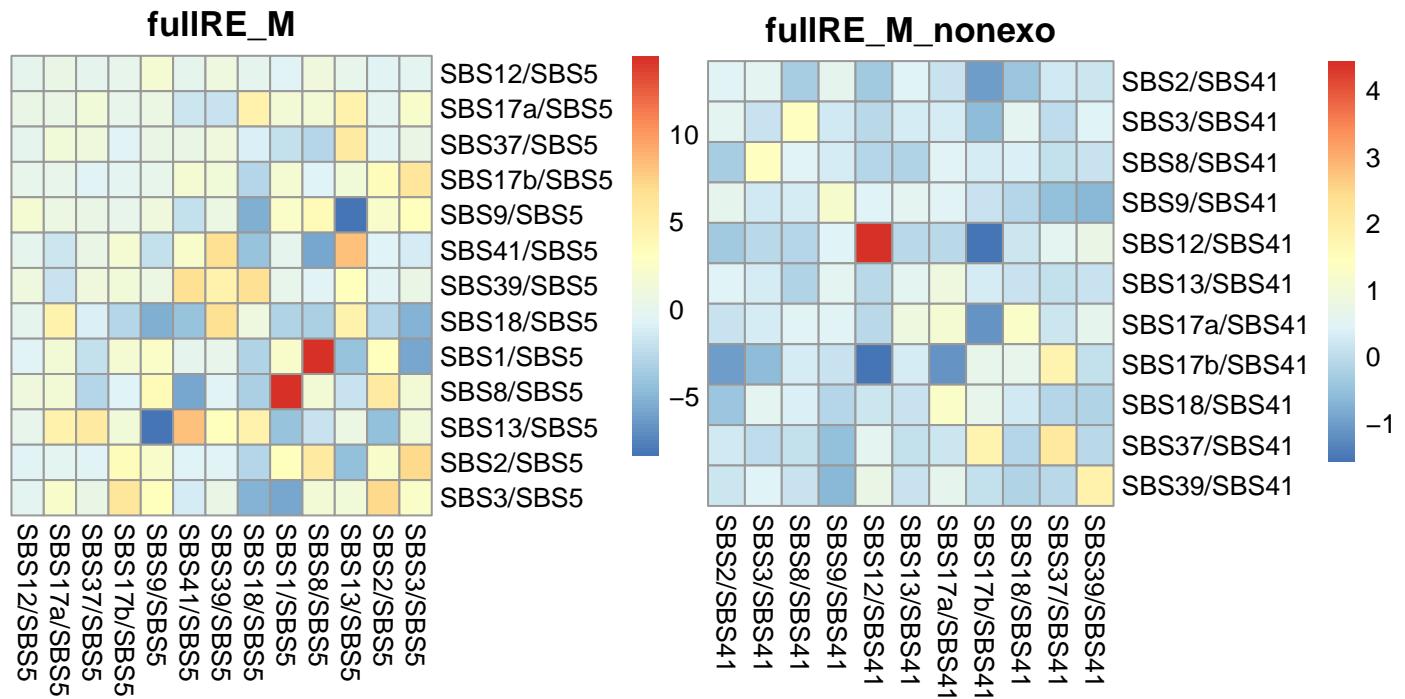
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma***(1/2) has now been replaced by (as we
## had before sometime in November) sigma

```

We use the results from the diagonal single lambda DM to test for differential abundance, giving a p-value of  $7.748574 \times 10^{-12}$ .

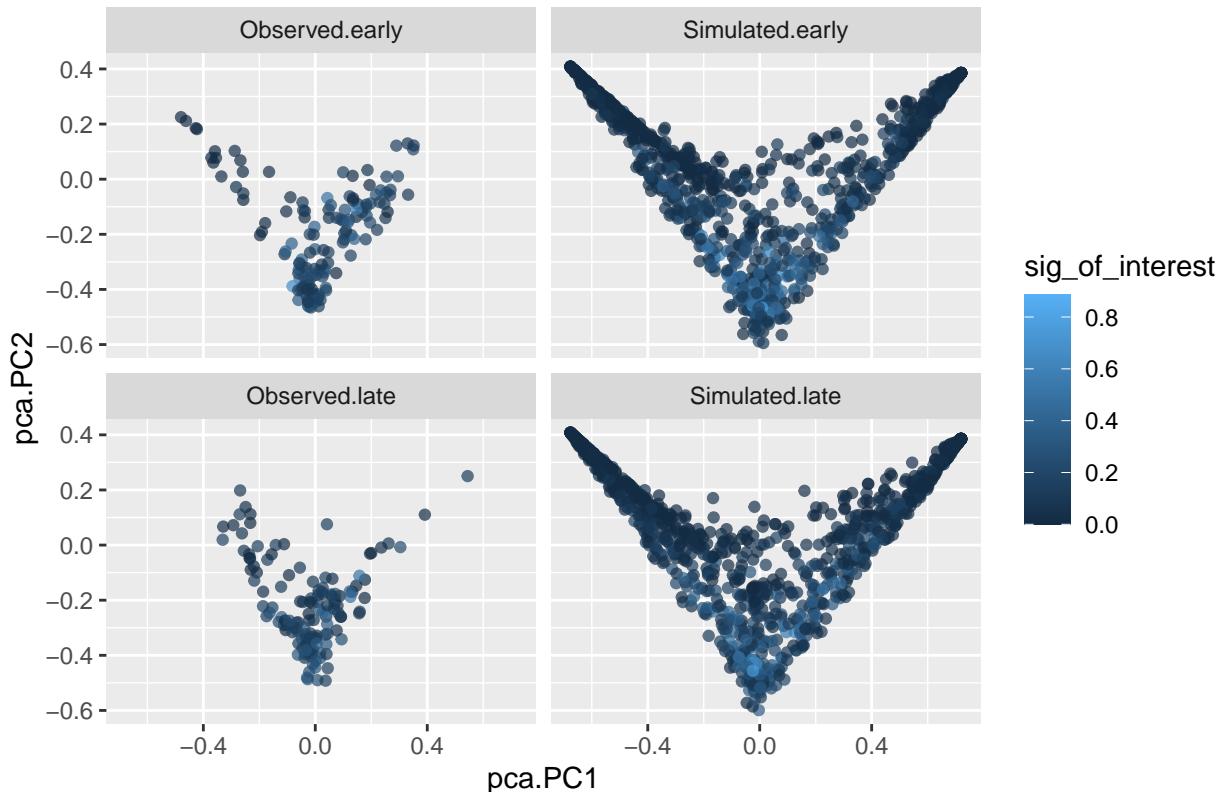
Covariance matrices



Simulation under inferred data

```
## [1] 136
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

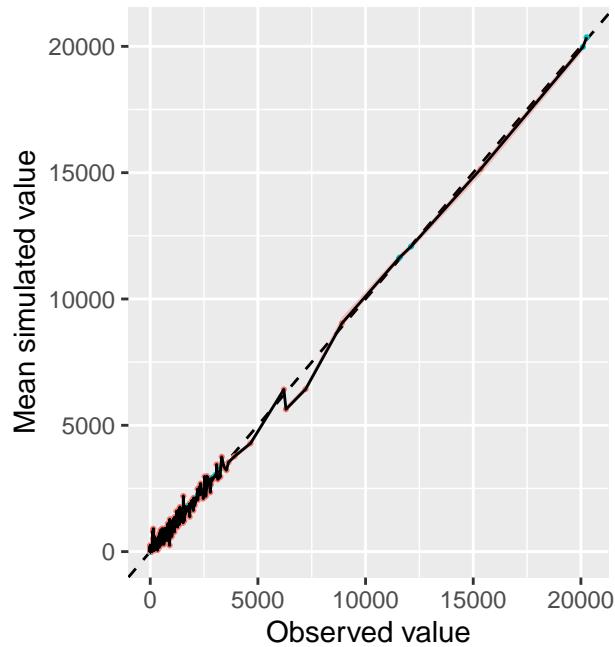
## Simulation of Breast Adenocarcinoma samples



## Ranked plot for coverage

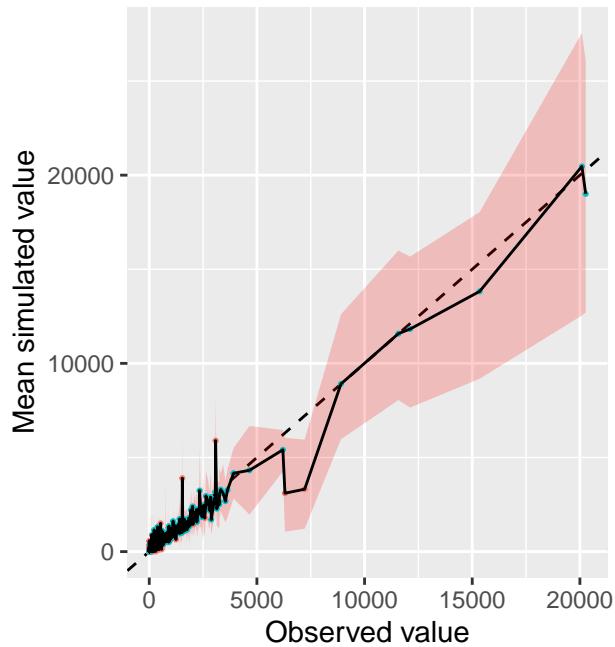
```
ct <- "Breast-AdenoCA"
integer_overdispersion_param_DMSL <- 1
obj_Breast_AdenoCA_nonexo <- give_subset_sigs_TMBobj(obj_Breast_AdenoCA, sigs_to_remove = nonexogenous$V1)
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_Breast_AdenoCA_nonexo,
print_plot = F, nreps = 20, model = "M")), function(i){
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )}[[1]],
data_object = obj_Breast_AdenoCA_nonexo,
loglog = F, title = 'Breast_AdenoCA_nonexo (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_Breast_AdenoCA_nonexo,
print_plot = F, nreps = 20, model = "DMSL", integer_overdispersion_param = integer_overdispersion_param_DMSL <- 1)),
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )}[[1]],
data_object = obj_Breast_AdenoCA_nonexo,
loglog = F, title = 'Breast_AdenoCA_nonexo (DMSL)'), ncol=2)
```

Breast\_AdenoCA\_nonexo (M)  
FALSE:2410; TRUE:854



col ● FALSE ● TRUE

Breast\_AdenoCA\_nonexo (DMS)  
FALSE:1348; TRUE:1916



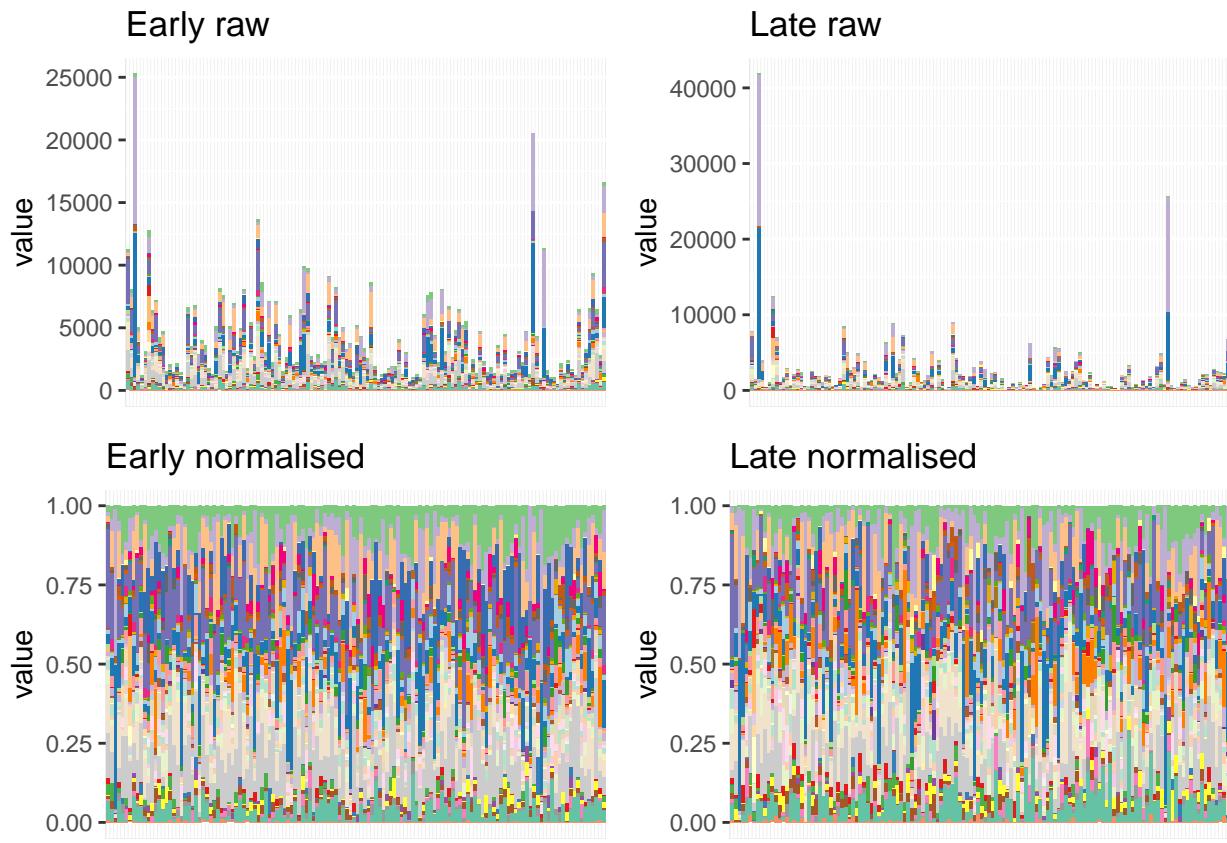
col ● FALSE ● TRUE

#### Signatures from mutSigExtractor

```
obj_Breast_AdenoCA_mutSigExtractor <- load_PCAWG(ct = "Breast-AdenoCA",
                                                    typedata = "signaturesmutSigExtractor",
                                                    path_to_data = "../data/")

## [1] 136
give_barplot_from_obj(obj = obj_Breast_AdenoCA_mutSigExtractor, legend_on = FALSE)

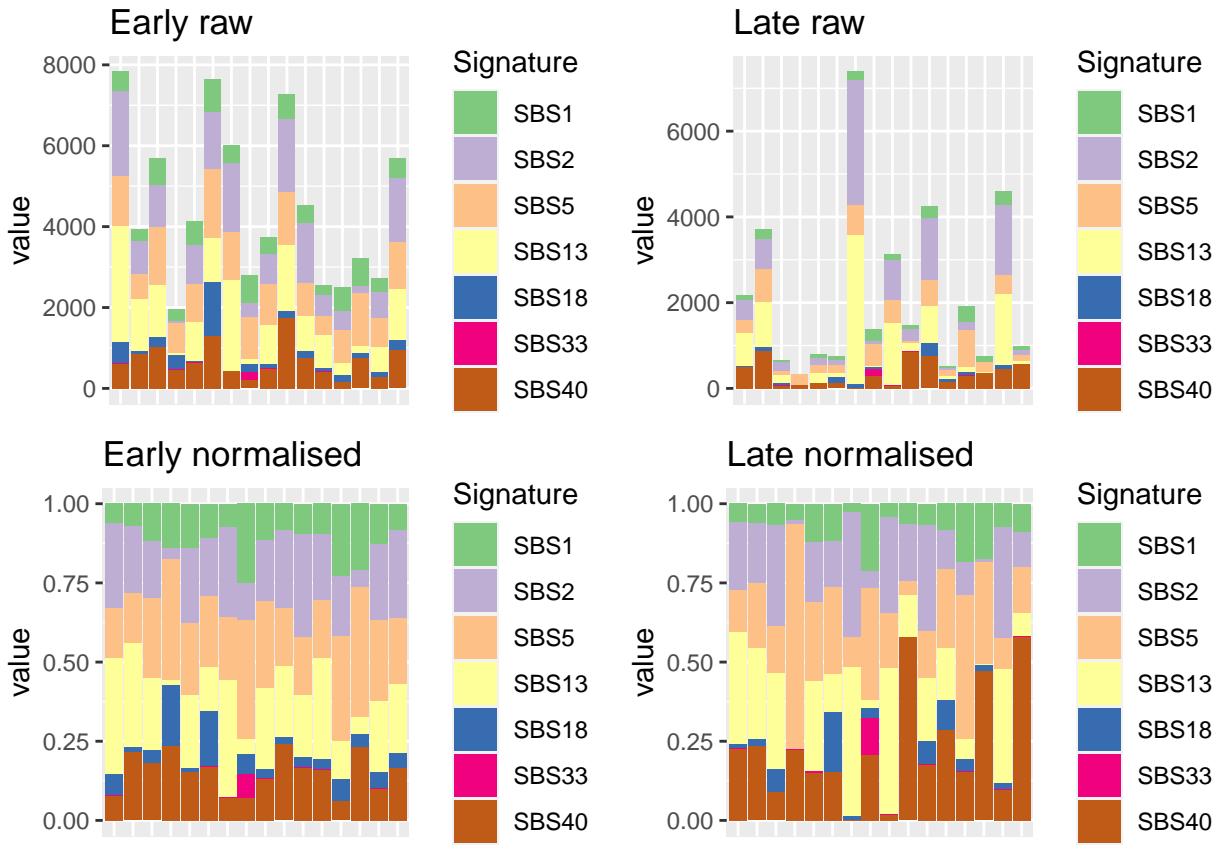
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
## Creating plot... it might take some time if the data are large. Number of samples: 136
```



## Cervix-SCC

### Barplot and general statistics

```
## [1] 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
```



The number of samples and signatures is:

```
## [1] 32 7
```

The signatures are:

```
## [1] "SBS1"  "SBS2"  "SBS5"  "SBS13" "SBS18" "SBS33" "SBS40"
```

#### Convergence table

```
##          value          L2          L1
## 1 Cervix-SCC hessian_positivedefinite_bool diagRE_M
## 2 Cervix-SCC hessian_nonpositivedefinite_bool fullRE_M
## 3 Cervix-SCC hessian_positivedefinite_bool diagRE_DMDL
## 4 Cervix-SCC hessian_nonpositivedefinite_bool fullRE_halfDM
## 5 Cervix-SCC hessian_nonpositivedefinite_bool fullRE_DMDL
## 6 Cervix-SCC hessian_positivedefinite_bool diagRE_DMSL
## 7 Cervix-SCC hessian_positivedefinite_bool sparseRE_DMSL
## 8 Cervix-SCC hessian_nonpositivedefinite_bool fullRE_DMSL
## 9 Cervix-SCC hessian_nonpositivedefinite_bool fullRE_DMSL_SBS1
## 10 Cervix-SCC hessian_positivedefinite_bool fullRE_M_nonexo
## 11 Cervix-SCC hessian_positivedefinite_bool diagRE_DMSL_nonexo
## 12 Cervix-SCC hessian_positivedefinite_bool sparseRE_DMSL_nonexo
## 13 Cervix-SCC hessian_positivedefinite_bool fullRE_DMSL_nonexo
## 14 Cervix-SCC hessian_positivedefinite_bool fullRE_DMDL_nonexo
## 15 Cervix-SCC hessian_positivedefinite_bool fullRE_DMDL_sortednonexo
```

## Potentially problematic signatures

SBS33 is a potentially problematic signature, being 0 in 81.2% of cases and with an overall exposure of 0.4%.

```
colSums(obj_Cervix_SCC$Y == 0)/nrow(obj_Cervix_SCC$Y)
```

```
##      SBS1      SBS2      SBS5     SBS13     SBS18     SBS33     SBS40
## 0.00000 0.00000 0.00000 0.03125 0.15625 0.81250 0.03125
colSums(obj_Cervix_SCC$Y)/sum(obj_Cervix_SCC$Y)

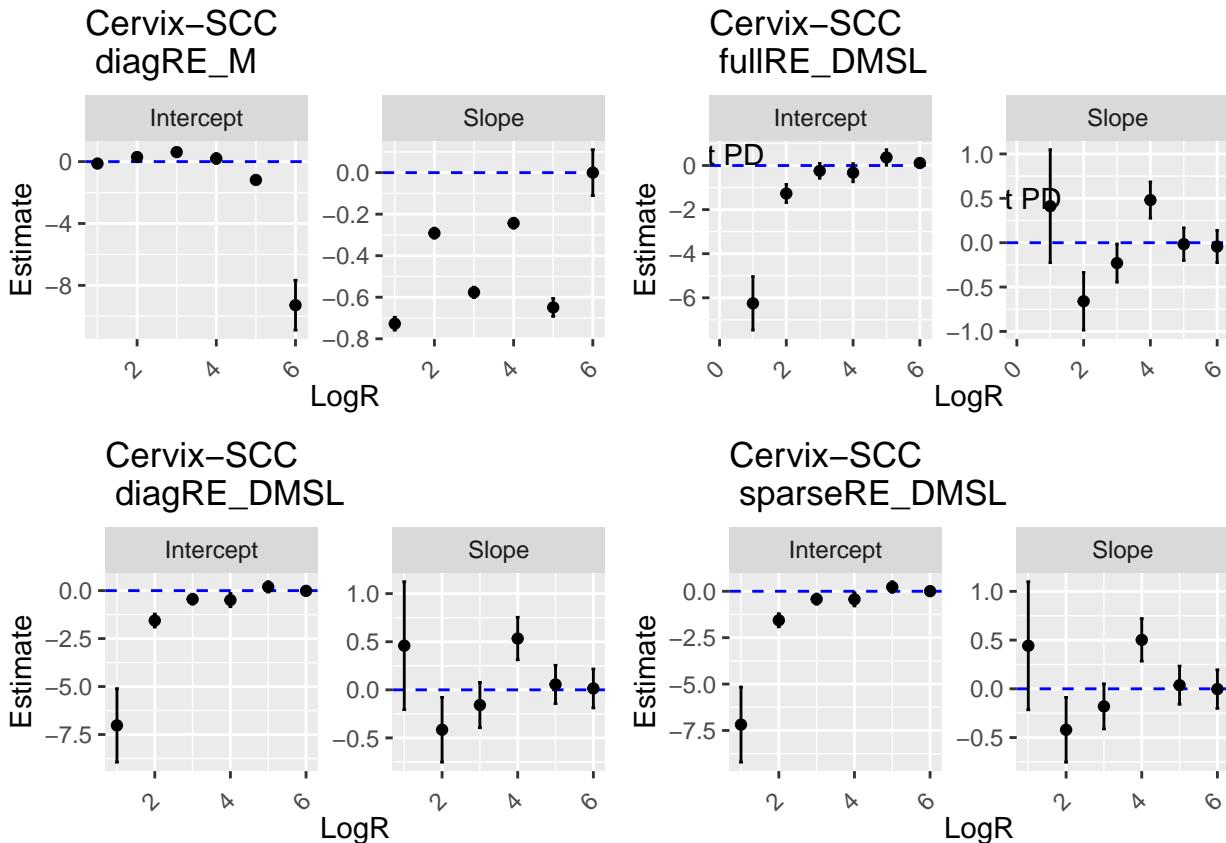
##      SBS1      SBS2      SBS5     SBS13     SBS18     SBS33
## 0.099164517 0.235000561 0.211562185 0.250439236 0.046577698 0.003560615
##      SBS40
## 0.153695189
```

## Betas

```
ct <- "Cervix-SCC"

grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
             plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
             plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
             plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)

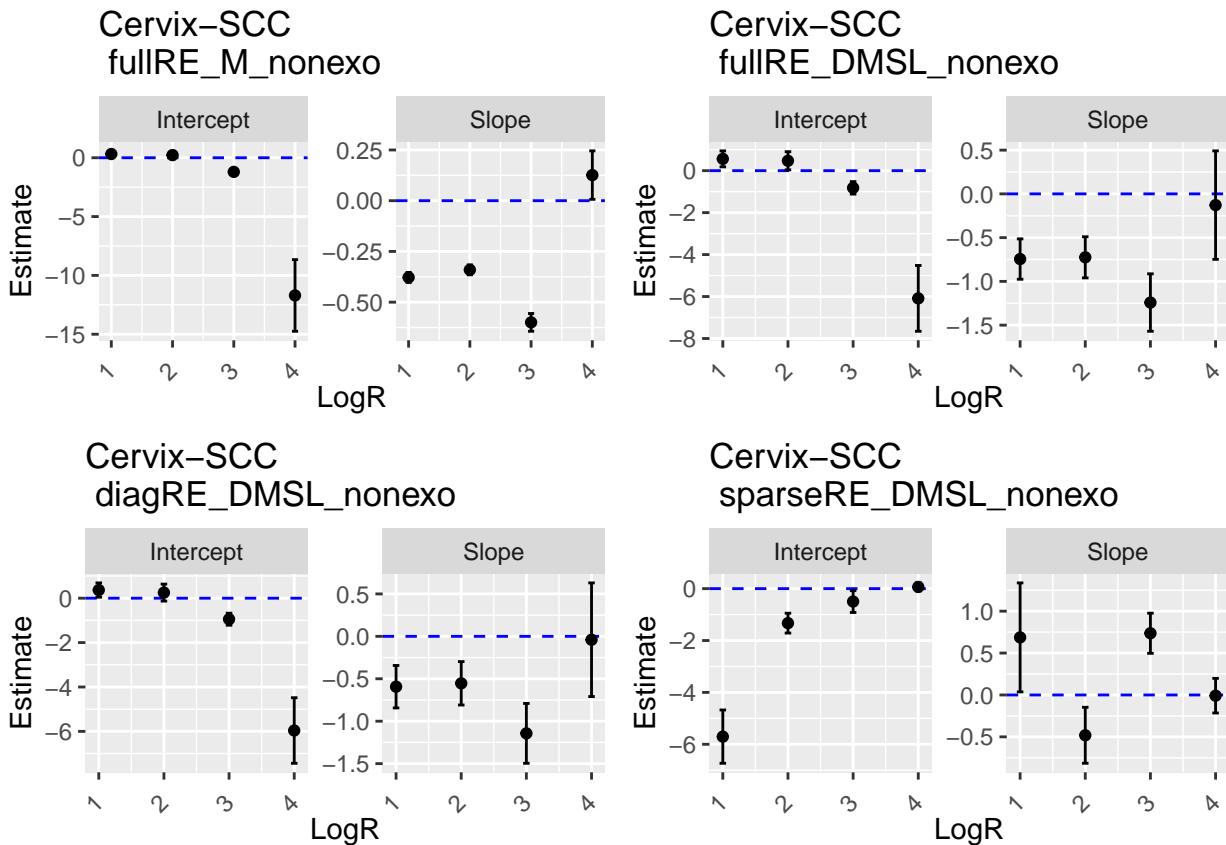
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```



```

grid.arrange(
  plot_betas(fullRE_M_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_M_nonexo')),
  plot_betas(fullRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL_nonexo')),
  plot_betas(diagRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL_nonexo')),
  plot_betas(sparseRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL_nonexo')), nrow=2)

```



```

## Warning in select_slope_2(which(names(i$par.fixed) == "beta"), verbatim
## = verbatim): As per 27 August it seems clear that this version, and not
## <select_slope>, is correct

```

```

## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

```

```

## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma**((1/2) has now been replaced by (as we
## had before sometime in November) sigma

```

We use the results from the fullRE single lambda DM to test for differential abundance, giving a p-value of  $3.8923434 \times 10^{-5}$ .

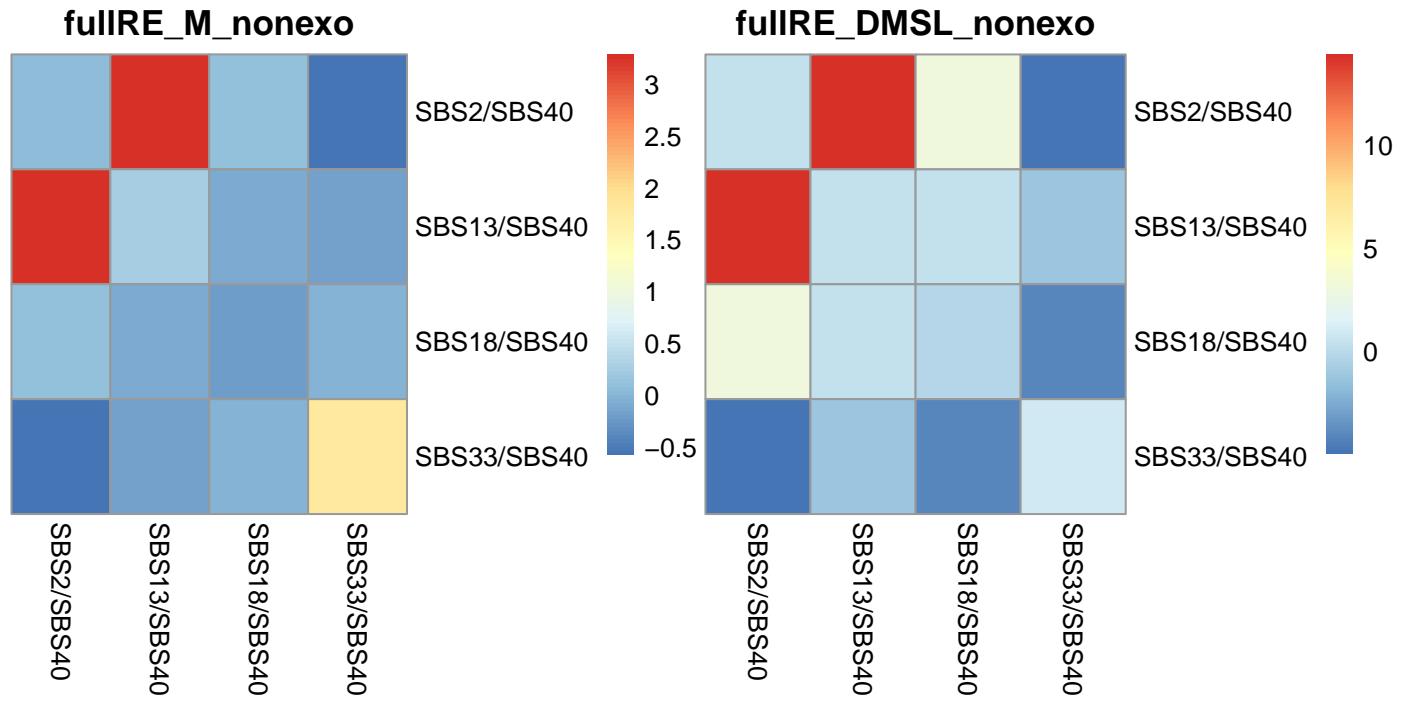
### Covariance matrices

```

# ct <- "Bone-Osteosarc"
# additional_sortedM <- list()
# additional_sortedDM <- list()

```

```
# additional_sortedM[[ct]] <- sortedM
# additional_sortedDM[[ct]] <- sortedDM
```



#### Simulation under inferred data

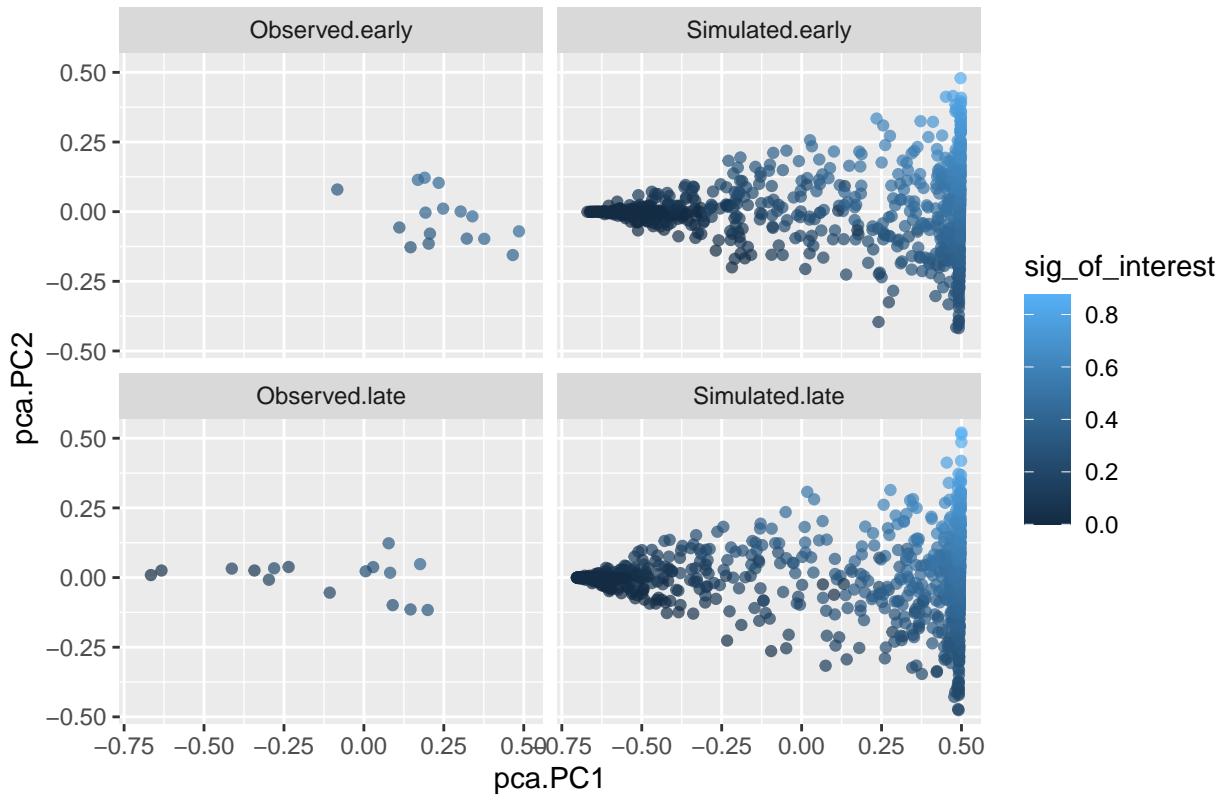
```
unique(nonexogenous$V1)

## [1] "SBS1"   "SBS4"   "SBS5"   "SBS7a"  "SBS7b"  "SBS7c"  "SBS7d"  "SBS11"  "SBS29"
## [10] "SBS31"  "SBS32"  "SBS35"  "SBS87"  "SBS92"  "SBS27"  "SBS43"  "SBS45"  "SBS46"
## [19] "SBS47"  "SBS48"  "SBS49"  "SBS50"  "SBS51"  "SBS52"  "SBS53"  "SBS54"  "SBS55"
## [28] "SBS56"  "SBS57"  "SBS58"  "SBS59"  "SBS60"

## [1] 16

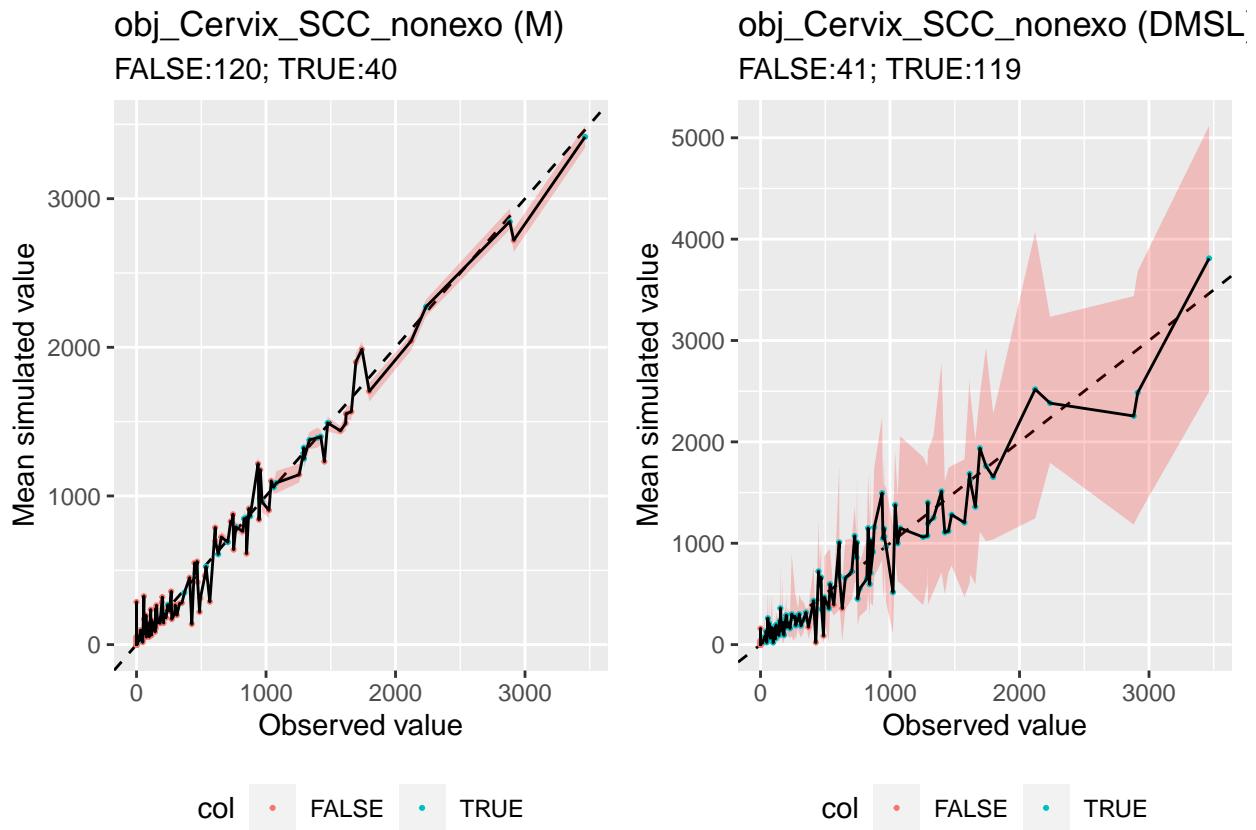
## Warning in mvtnorm:::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

## Simulation of Cervix SCC samples



### Ranked plot for coverage

```
ct <- "Cervix-SCC"
integer_overdispersion_param_DMSL <- 1
obj_Cervix_SCC_nonexo <- give_subset_sigs_TMBobj(obj_Cervix_SCC, sigs_to_remove = nonexogenous$V1)
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
                                                       data_object = obj_Cervix_SCC_nonexo,
                                                       print_plot = F, nreps = 20, model = "M")),
                                               function(i){lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
                                                                                           rank_number=1:length(j)) )}[[1]],
                                              data_object = obj_Cervix_SCC_nonexo,
                                              loglog = F, title = 'obj_Cervix_SCC_nonexo (M)'), 
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
                                                       data_object = obj_Cervix_SCC_nonexo,
                                                       print_plot = F, nreps = 20, model = "DMSL",
                                                       integer_overdispersion_param = integer_overdispersion_param_DMSL),
                                               function(i){lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
                                                                                           rank_number=1:length(j)) )}[[1]],
                                              data_object = obj_Cervix_SCC_nonexo,
                                              loglog = F, title = 'obj_Cervix_SCC_nonexo (DMSL)'), ncol=2)
```



#### Signatures from mutSigExtractor

```
obj_Cervix_SCC_mutSigExtractor <- load_PCAWG(ct = "Cervix-SCC", typedata = "signaturesmutSigExtractor",
                                              path_to_data = "../..../data/")

## [1] 16

give_barplot_from_obj(obj = obj_Cervix_SCC_mutSigExtractor, legend_on = TRUE)

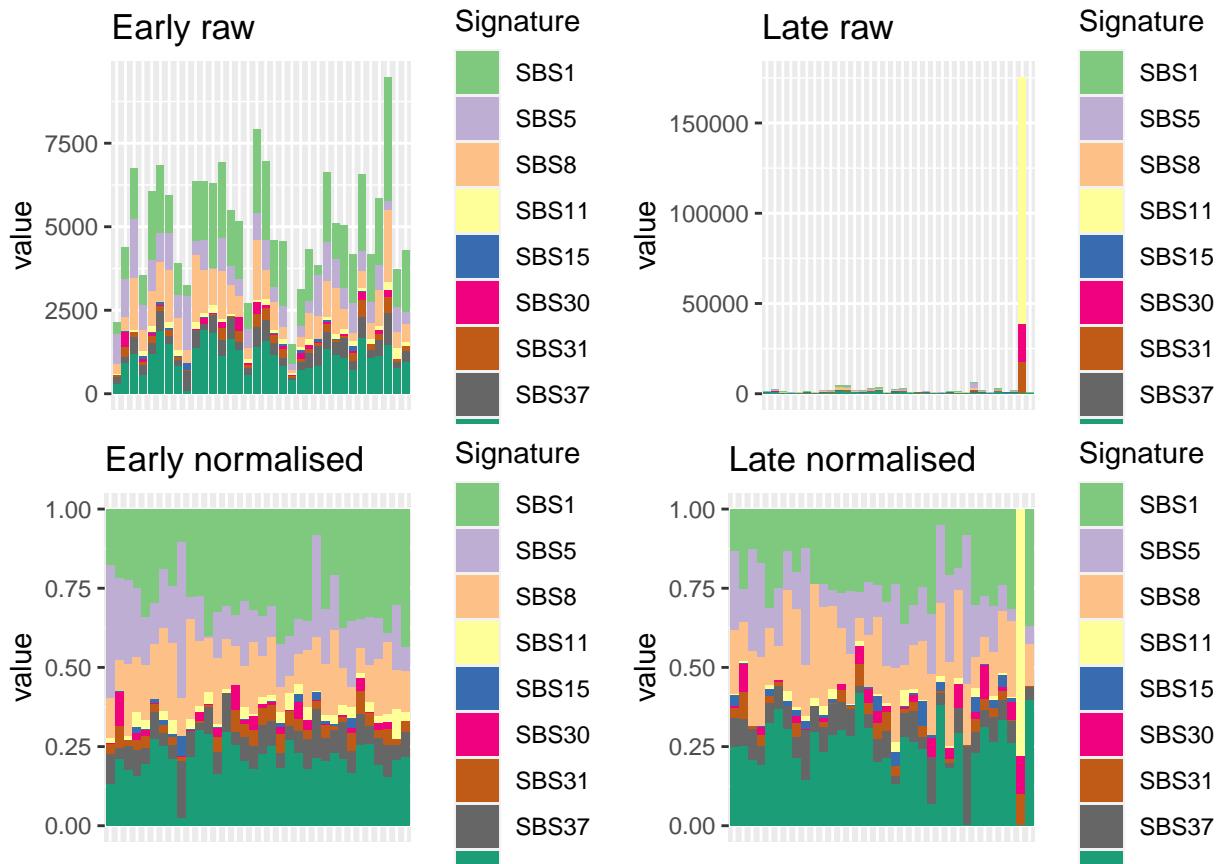
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
```



## CNS-GBM

### Barplot and general statistics

```
## [1] 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
```



The number of samples and signatures is:

```
## [1] 68 9
```

The signatures are:

```
## [1] "SBS1"  "SBS5"  "SBS8"  "SBS11" "SBS15" "SBS30" "SBS31" "SBS37" "SBS40"
```

### Convergence table

We only have converged results for the multinomial with full RE, and the DM with a single lambda (diag and sparse RE). It is the same for nonexogenous signatures.

	value	L2	L1
## 1	CNS-GBM hessian_positivedefinite_bool		diagRE_M
## 2	CNS-GBM hessian_positivedefinite_bool		fullRE_M
## 3	CNS-GBM hessian_nonpositivedefinite_bool		diagRE_DMDL
## 4	CNS-GBM hessian_nonpositivedefinite_bool		fullRE_halfDM
## 5	CNS-GBM hessian_nonpositivedefinite_bool		fullRE_DMDL
## 6	CNS-GBM hessian_positivedefinite_bool		diagRE_DMSL
## 7	CNS-GBM hessian_positivedefinite_bool		sparseRE_DMSL
## 8	CNS-GBM hessian_nonpositivedefinite_bool		fullRE_DMSL
## 9	CNS-GBM hessian_nonpositivedefinite_bool		fullRE_DMSL_SBS1
## 10	CNS-GBM hessian_positivedefinite_bool		fullRE_M_nonexo
## 11	CNS-GBM hessian_positivedefinite_bool		diagRE_DMSL_nonexo
## 12	CNS-GBM hessian_positivedefinite_bool		sparseRE_DMSL_nonexo

```

## 13 CNS-GBM hessian_nonpositivedefinite_bool      fullRE_DMSL_nonexo
## 14 CNS-GBM hessian_nonpositivedefinite_bool      fullRE_DMDL_nonexo
## 15 CNS-GBM                                     Timeout fullRE_DMDL_sortednonexo

```

### Re-running of fitting

Using fullRE\_M\_nonexo to fit fullRE\_DMSL\_nonexo HEREEE

If we use the values of the fullRE M exo as initial values for the fullRE DMSL exo do converge:

```
## [1] TRUE
```

### Potentially problematic signatures

We notice that there are no truly problematic signatures (SBS15 has the most zeros; 50%).

```
colSums(obj_CNS_GBM$Y == 0)/nrow(obj_CNS_GBM$Y)
```

```

##      SBS1      SBS5      SBS8      SBS11     SBS15      SBS30      SBS31
## 0.01470588 0.02941176 0.01470588 0.20588235 0.50000000 0.33823529 0.13235294
##      SBS37      SBS40
## 0.01470588 0.02941176

```

```
colSums(obj_CNS_GBM$Y)/sum(obj_CNS_GBM$Y)
```

```

##      SBS1      SBS5      SBS8      SBS11     SBS15      SBS30
## 0.164856854 0.087757118 0.103223676 0.345294365 0.004258098 0.060917020
##      SBS31      SBS37      SBS40
## 0.060793210 0.046931329 0.125968329
additional_sortedMnonexo <- list()
additional_sortedDMSLnonexo <- list()

```

```

additional_sortedMnonexo[["CNS-GBM"]] <- sortedM_CNSGBM
additional_sortedDMSLnonexo[["CNS-GBM"]] <- sortedDM_CNSGBM

```

### Betas

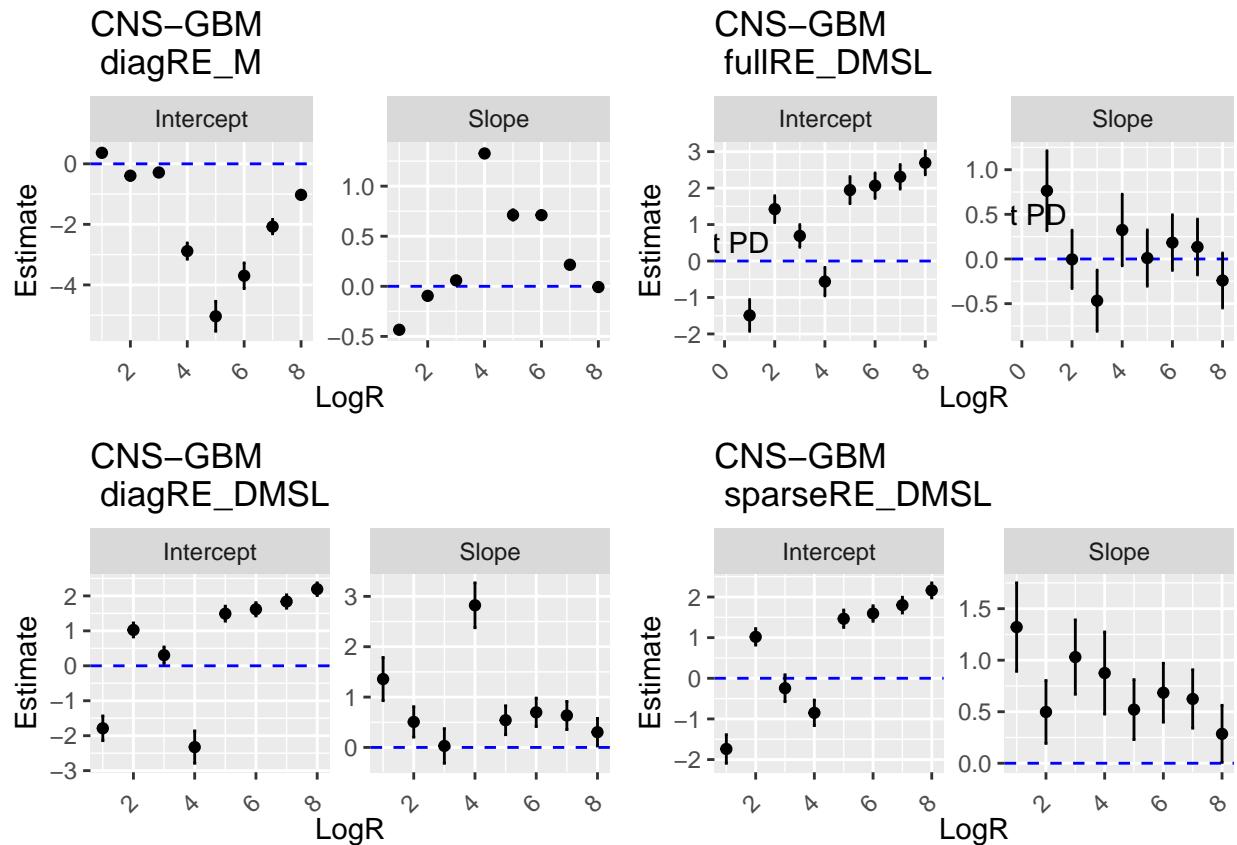
```

ct <- "CNS-GBM"

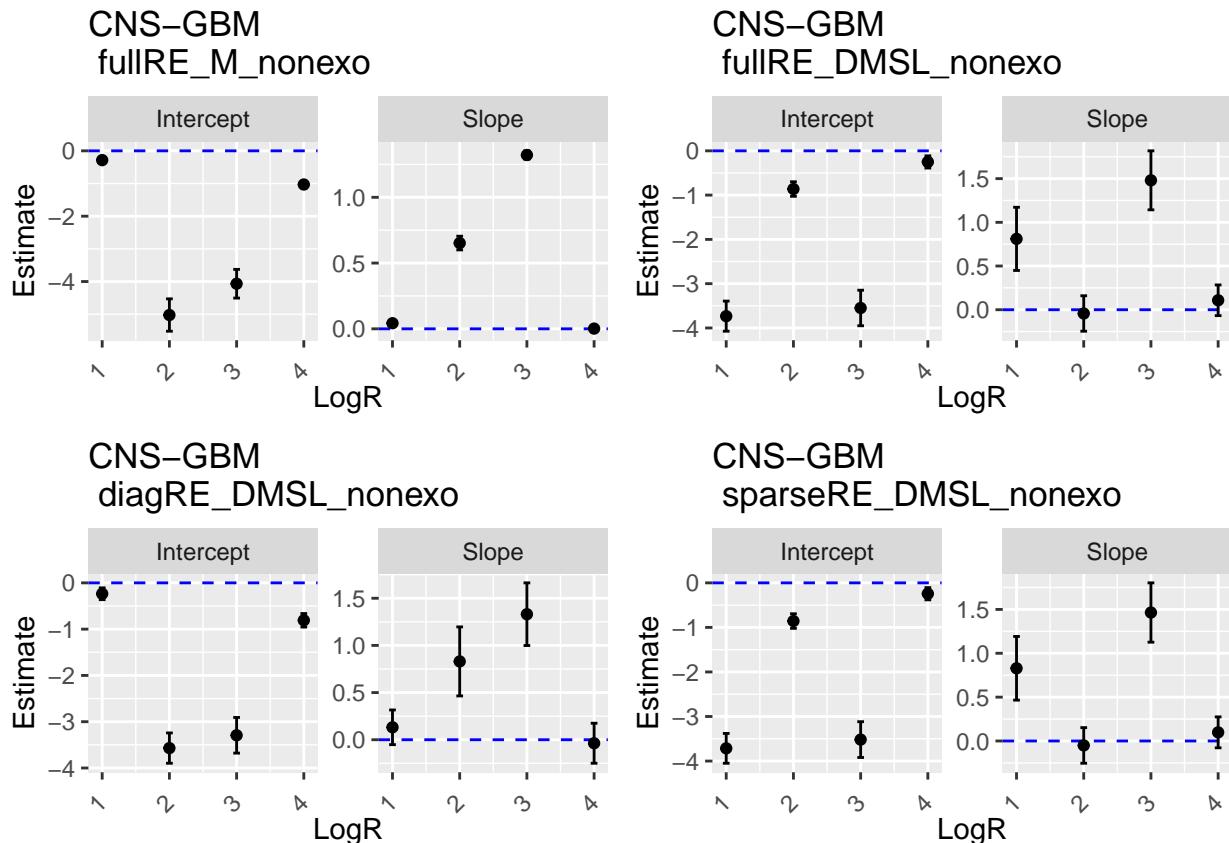
grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)

## Warning in sqrt(diag(object$cov.fixed)): NaNs produced

```



```
grid.arrange(
  plot_betas(fullRE_M_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_M_nonexo')),
  plot_betas(sortedDM_CNSGBM)+ggtitle(paste0(ct, '\n fullRE_DMSL_nonexo')),
  plot_betas(diagRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL_nonexo')),
  plot_betas(sparseRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL_nonexo')), nrow=2)
```



```

## Warning in select_slope_2(which(names(i$par.fixed) == "beta"), verbatim
## = verbatim): As per 27 August it seems clear that this version, and not
## <select_slope>, is correct

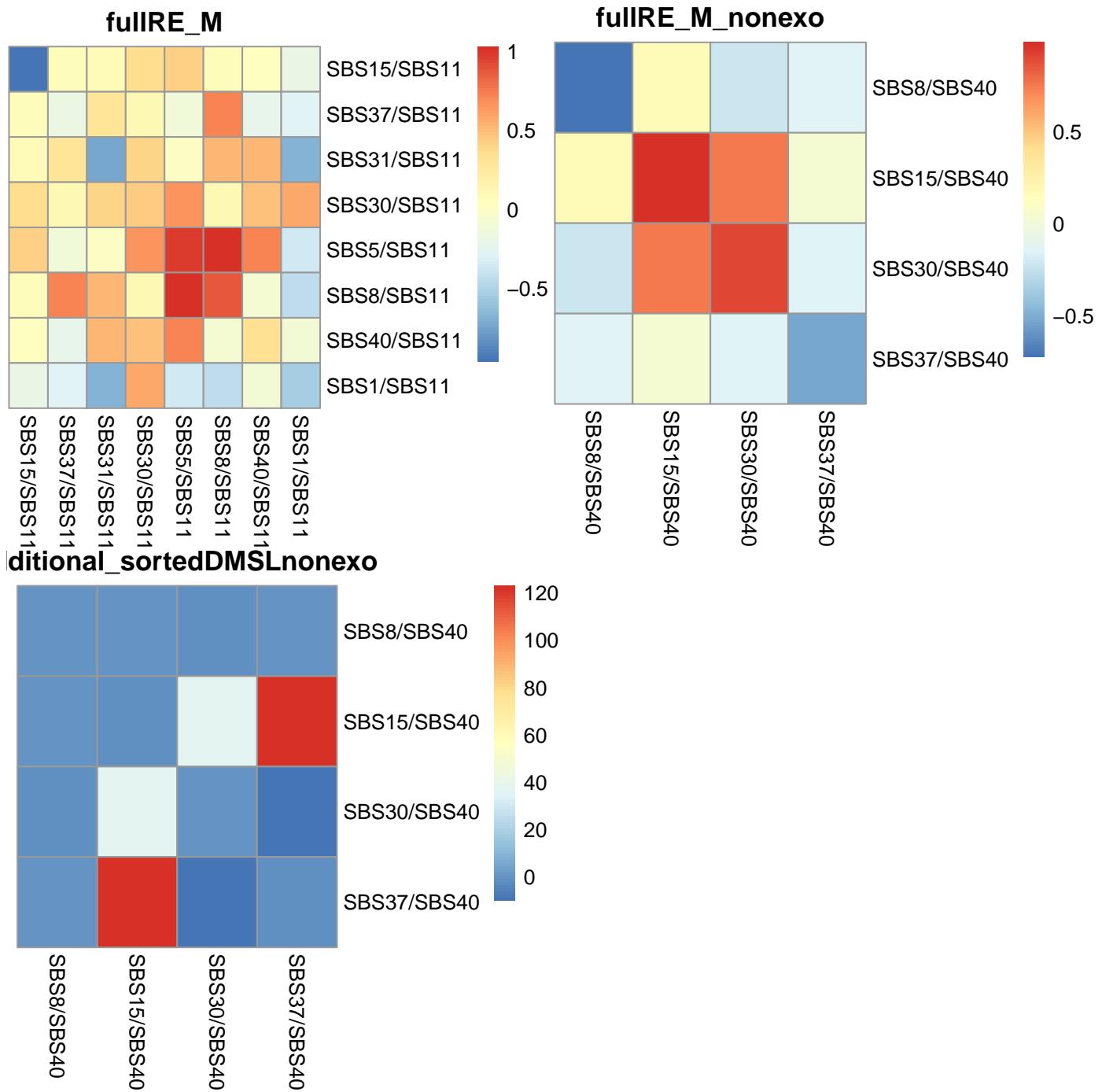
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma**((1/2) has now been replaced by (as we
## had before sometime in November) sigma

```

We use the results from the full RE single lambda DM to test for differential abundance, giving a p-value of  $6.6827492 \times 10^{-5}$ .

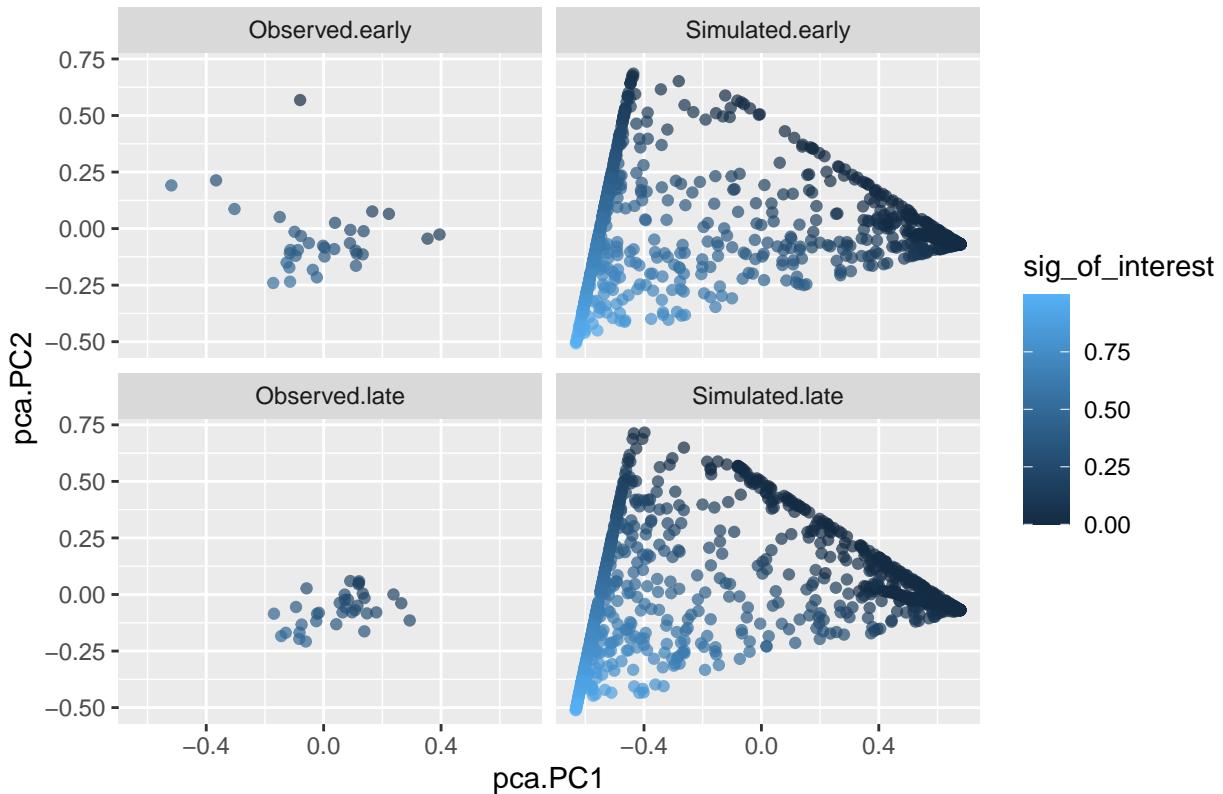
### Covariance matrices



### Simulation under inferred data

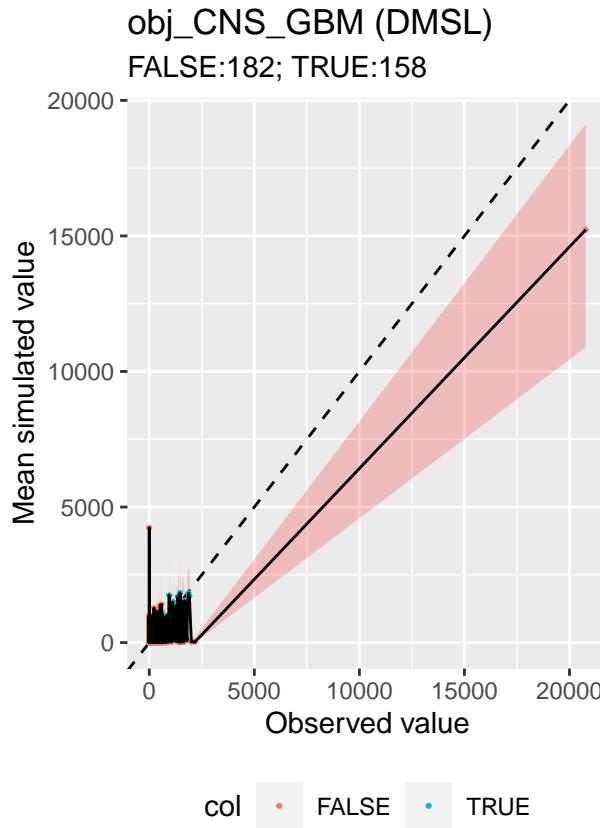
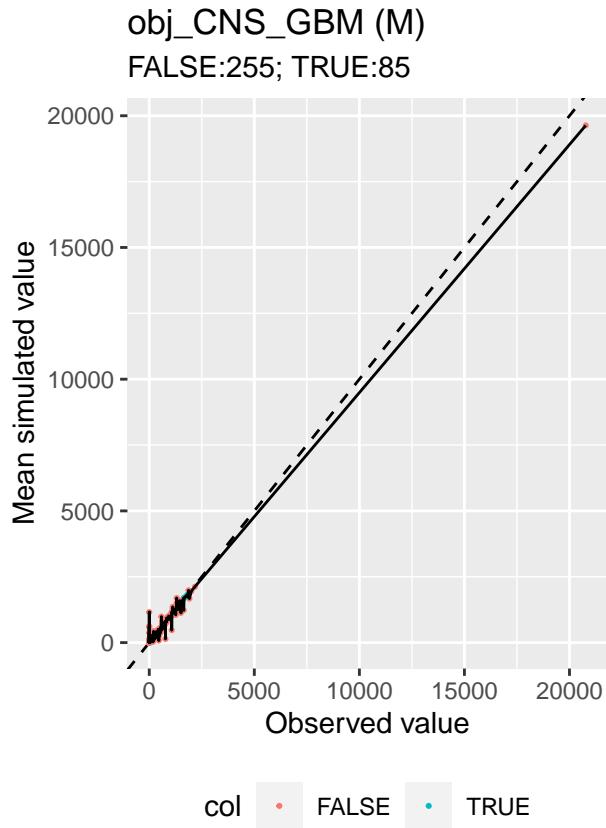
```
## [1] 34
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

## Simulation of CNS-GBM samples



## Ranked plot for coverage

```
ct <- "CNS-GBM"
integer_overdispersion_param_DMSL <- 1
obj_CNS_GBM_nonexo <- give_subset_sigs_TMBObj(obj_CNS_GBM, sigs_to_remove = nonexogenous$V1)
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = full,
data_object = obj_CNS_GBM_nonexo,
print_plot = F, nreps = 20, model = "M")), function(i){
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )}[[1]],
data_object = obj_CNS_GBM_nonexo,
loglog = F, title = 'obj_CNS_GBM (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = sortedDM_CNSGBM,
data_object = obj_CNS_GBM_nonexo,
print_plot = F, nreps = 20, model = "DMSL", integer_overdispersion_param = integer_overdispersion_param_DMSL),
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )}[[1]],
data_object = obj_CNS_GBM_nonexo,
loglog = F, title = 'obj_CNS_GBM (DMSL)'), ncol=2)
```



Surprisingly, the values for DMSL look even worse than the multinomial, for high values

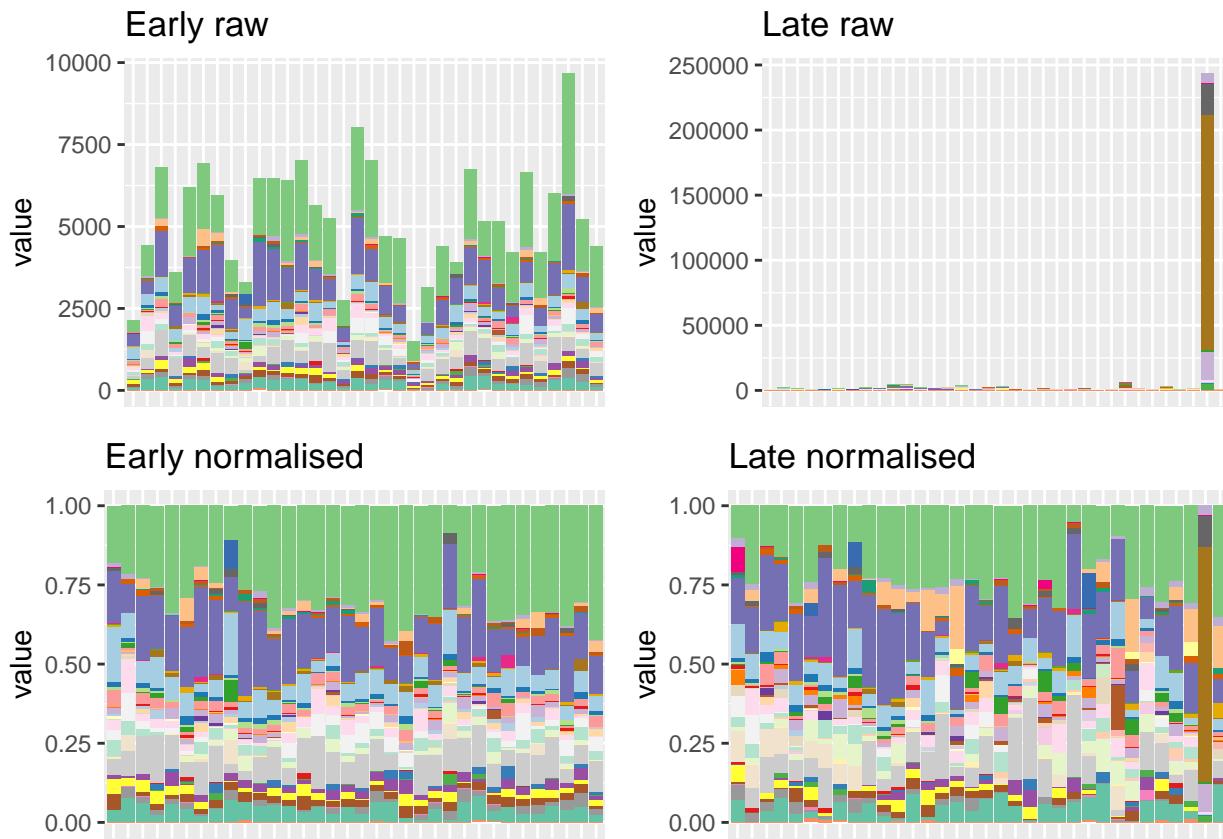
### Signatures from mutSigExtractor

The signatures from mutSigExtractor are a bit more chaotic:

```
obj_CNS_GBM_mutSigExtractor <- load_PCAWG(ct = ct, typedata = "signaturesmutSigExtractor",
                                             path_to_data = "../data/")

## [1] 34
give_barplot_from_obj(obj = obj_CNS_GBM_mutSigExtractor, legend_on = FALSE)

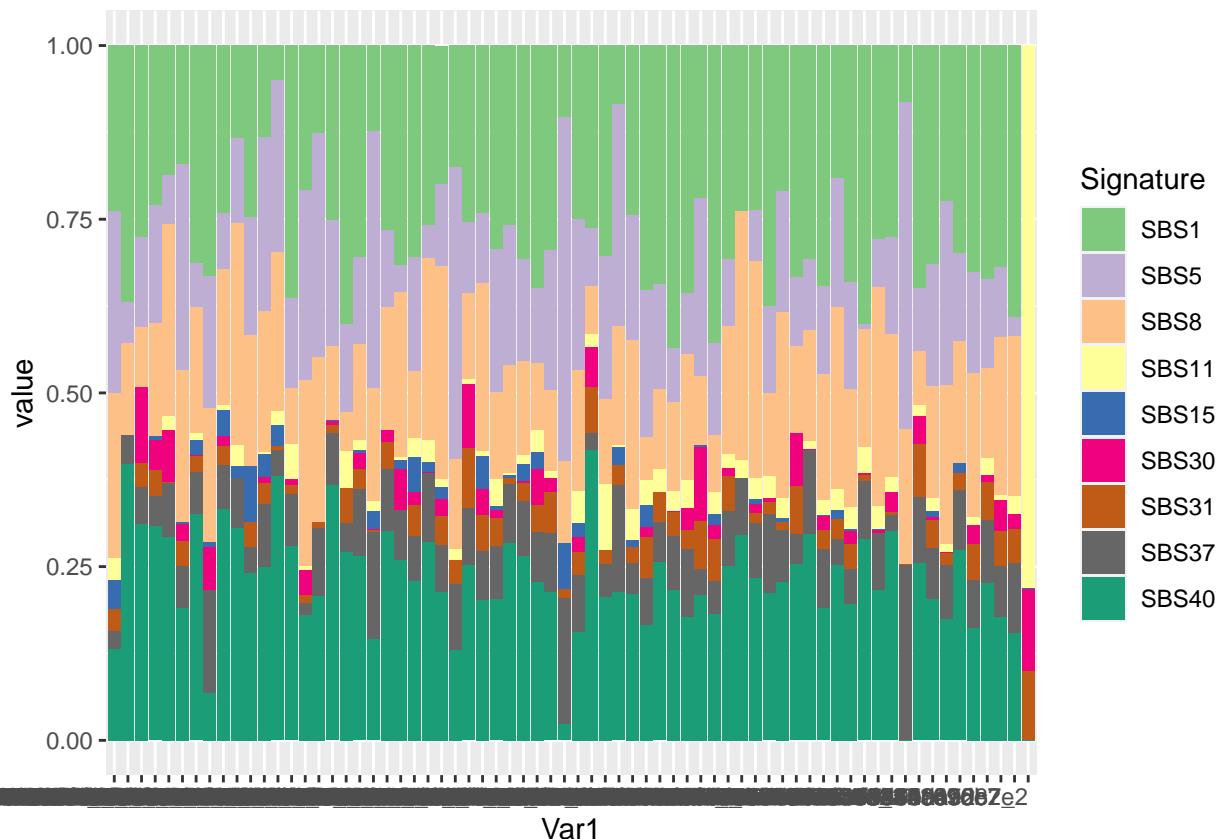
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
```



Exposures sorted by increasing number of mutations: there is no trend of signatures being associated with the number of mutations.

```
createBarplot(normalise_rw(non_duplicated_rows(obj_CNS_GBM$Y)),
              order_labels = names(sort(rowSums(non_duplicated_rows(obj_CNS_GBM$Y)),
                                         decreasing = F)))
```

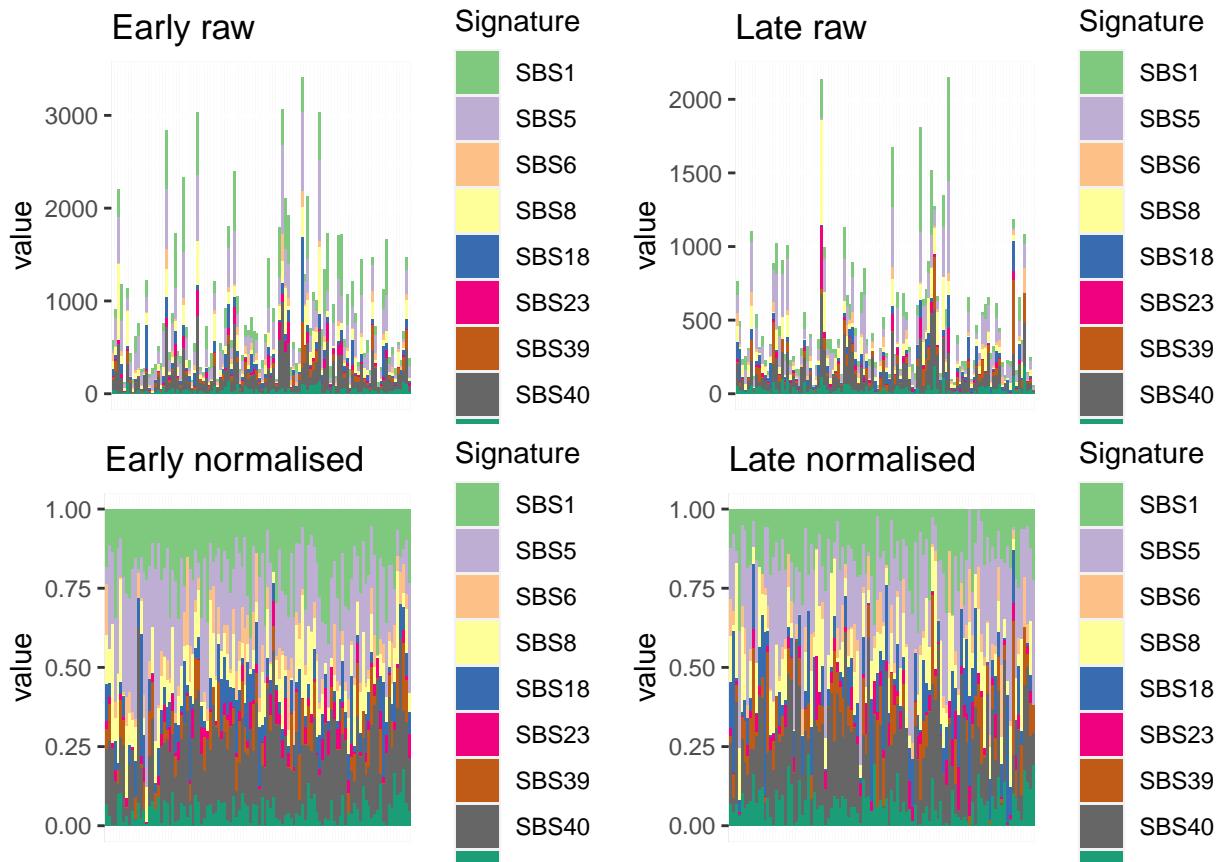
```
## Creating plot... it might take some time if the data are large. Number of samples: 68
```



## CNS-Medullo

### Barplot and general statistics

```
## [1] 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
```



The number of samples and signatures is:

```
## [1] 212 9
```

The signatures are:

```
## [1] "SBS1" "SBS5" "SBS6" "SBS8" "SBS18" "SBS23" "SBS39" "SBS40" "SBS46"
```

### Convergence table

Pretty much everything has converged in this case

##	value	L2	L1
## 1	CNS-Medullo	hessian_positivedefinite_bool	diagRE_M
## 2	CNS-Medullo	hessian_positivedefinite_bool	fullRE_M
## 3	CNS-Medullo	hessian_positivedefinite_bool	diagRE_DMDL
## 4	CNS-Medullo	hessian_nonpositivedefinite_bool	fullRE_halfDM
## 5	CNS-Medullo	hessian_nonpositivedefinite_bool	fullRE_DMDL
## 6	CNS-Medullo	hessian_positivedefinite_bool	diagRE_DMSL
## 7	CNS-Medullo	hessian_positivedefinite_bool	sparseRE_DMSL
## 8	CNS-Medullo	hessian_nonpositivedefinite_bool	fullRE_DMSL
## 9	CNS-Medullo	hessian_nonpositivedefinite_bool	fullRE_DMSL_SBS1
## 10	CNS-Medullo	hessian_positivedefinite_bool	fullRE_M_nonexo
## 11	CNS-Medullo	hessian_positivedefinite_bool	diagRE_DMSL_nonexo
## 12	CNS-Medullo	hessian_positivedefinite_bool	sparseRE_DMSL_nonexo
## 13	CNS-Medullo	hessian_positivedefinite_bool	fullRE_DMSL_nonexo

```
## 14 CNS-Medullo hessian_positivedefinite_bool fullRE_DMDL_nonexo
## 15 CNS-Medullo hessian_nonpositivedefinite_bool fullRE_DMDL_sortednonexo
```

As nonexo DMSL has already converged, we don't re-run anything.

### Potentially problematic signatures

We notice that there are no truly problematic signatures

```
colSums(obj_CNS_Medullo$Y == 0)/nrow(obj_CNS_Medullo$Y)
```

```
##      SBS1      SBS5      SBS6      SBS8      SBS18      SBS23
## 0.004716981 0.056603774 0.264150943 0.089622642 0.155660377 0.235849057
##      SBS39      SBS40      SBS46
## 0.353773585 0.066037736 0.099056604
```

```
colSums(obj_CNS_Medullo$Y)/sum(obj_CNS_Medullo$Y)
```

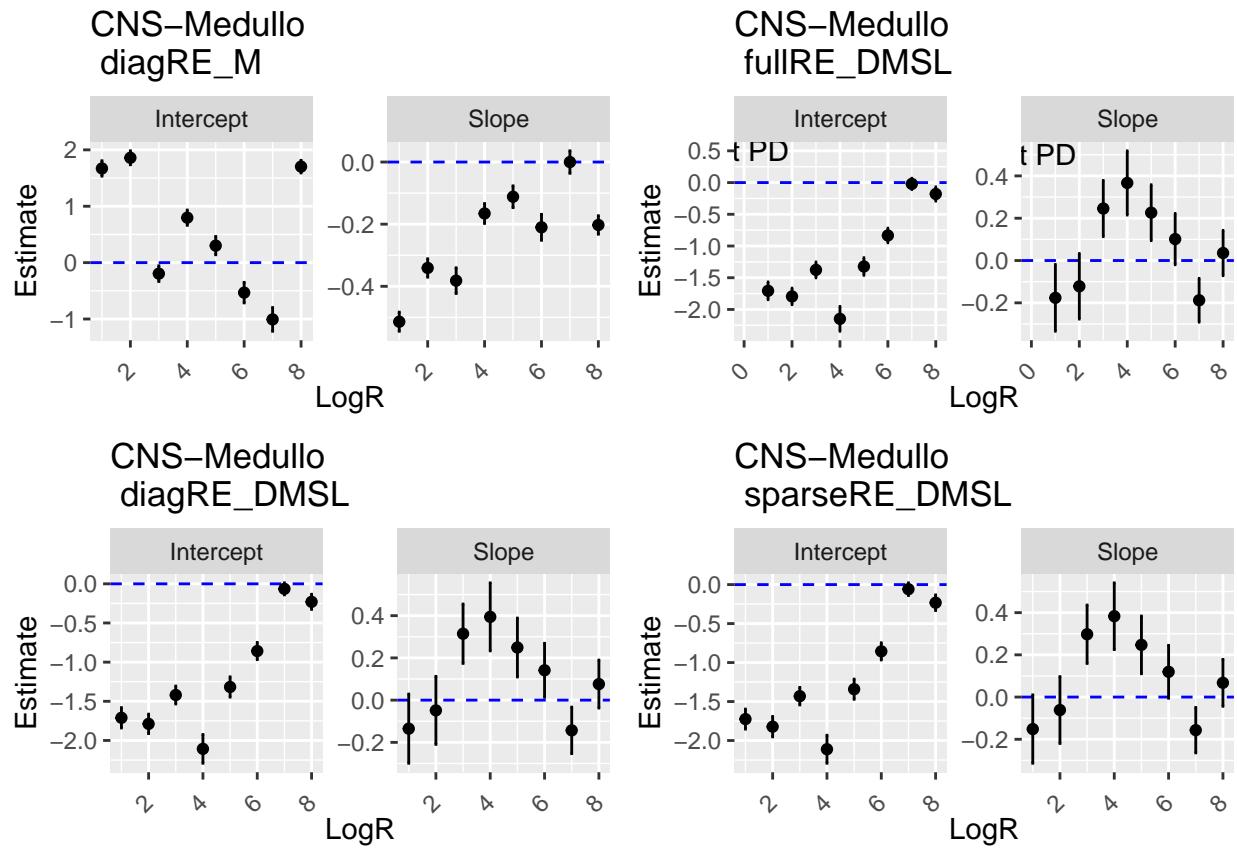
```
##      SBS1      SBS5      SBS6      SBS8      SBS18      SBS23      SBS39
## 0.19177483 0.22946904 0.03737123 0.11614418 0.07466844 0.03836035 0.05498025
##      SBS40      SBS46
## 0.21065558 0.04657610
```

### Betas

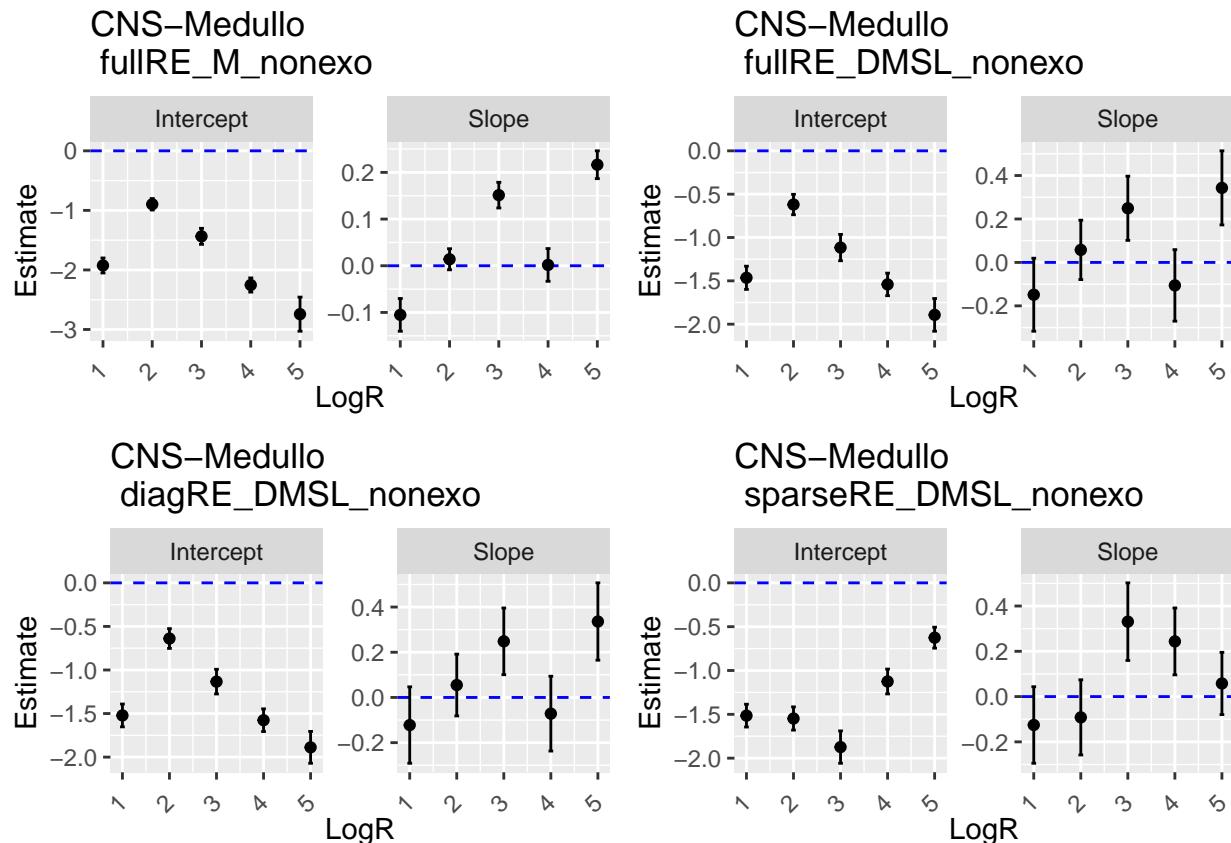
```
ct <- "CNS-Medullo"

grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)

## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```



```
grid.arrange(
  plot_betas(fullRE_M_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_M_nonexo')),
  plot_betas(fullRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL_nonexo')),
  plot_betas(diagRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL_nonexo')),
  plot_betas(sparseRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL_nonexo')), nrow=2)
```



```

## Warning in select_slope_2(which(names(i$par.fixed) == "beta"), verbatim
## = verbatim): As per 27 August it seems clear that this version, and not
## <select_slope>, is correct

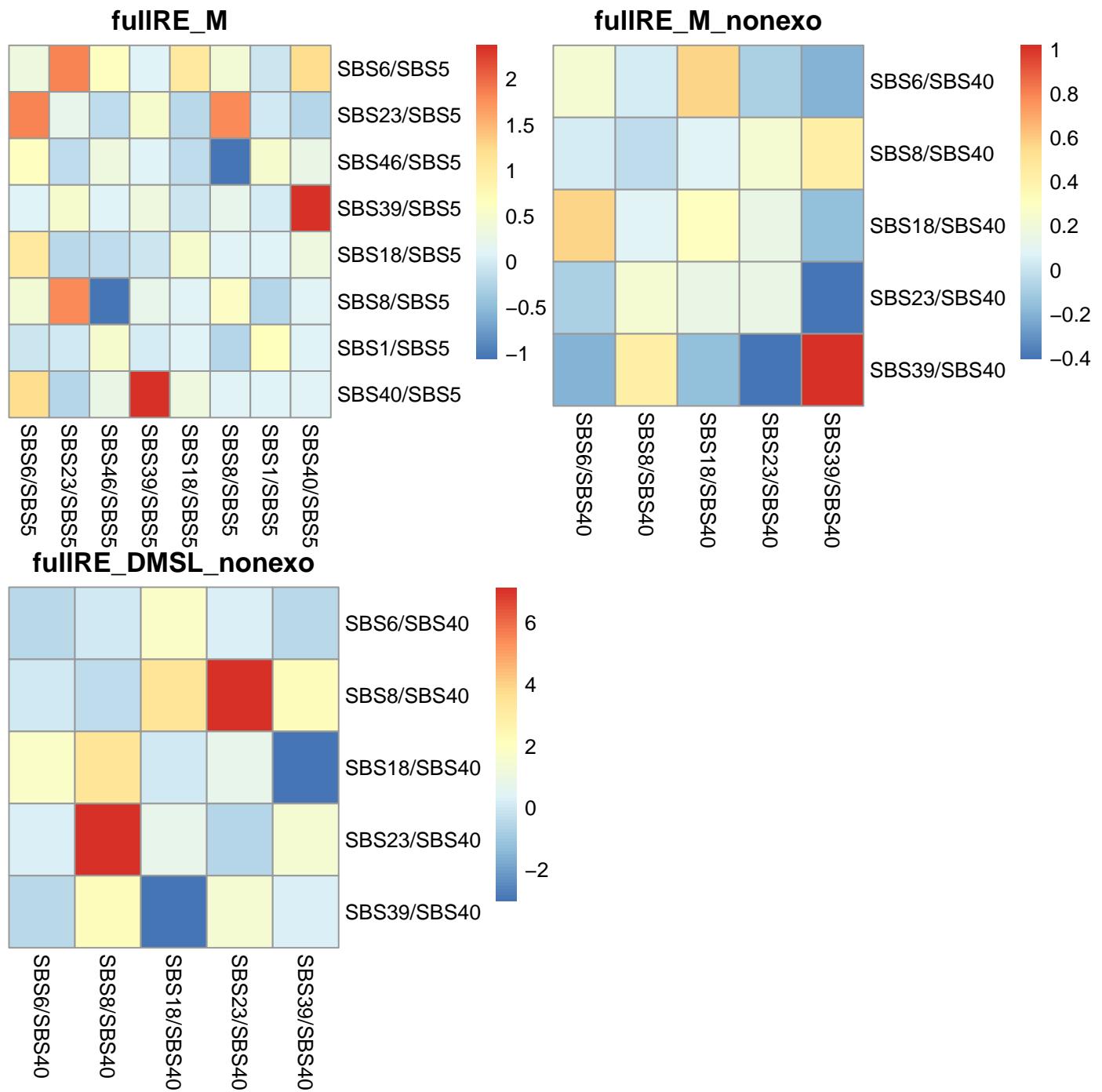
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma**((1/2) has now been replaced by (as we
## had before sometime in November) sigma

```

We use the results from the full RE single lambda DM to test for differential abundance, giving a p-value of 0.0677062.

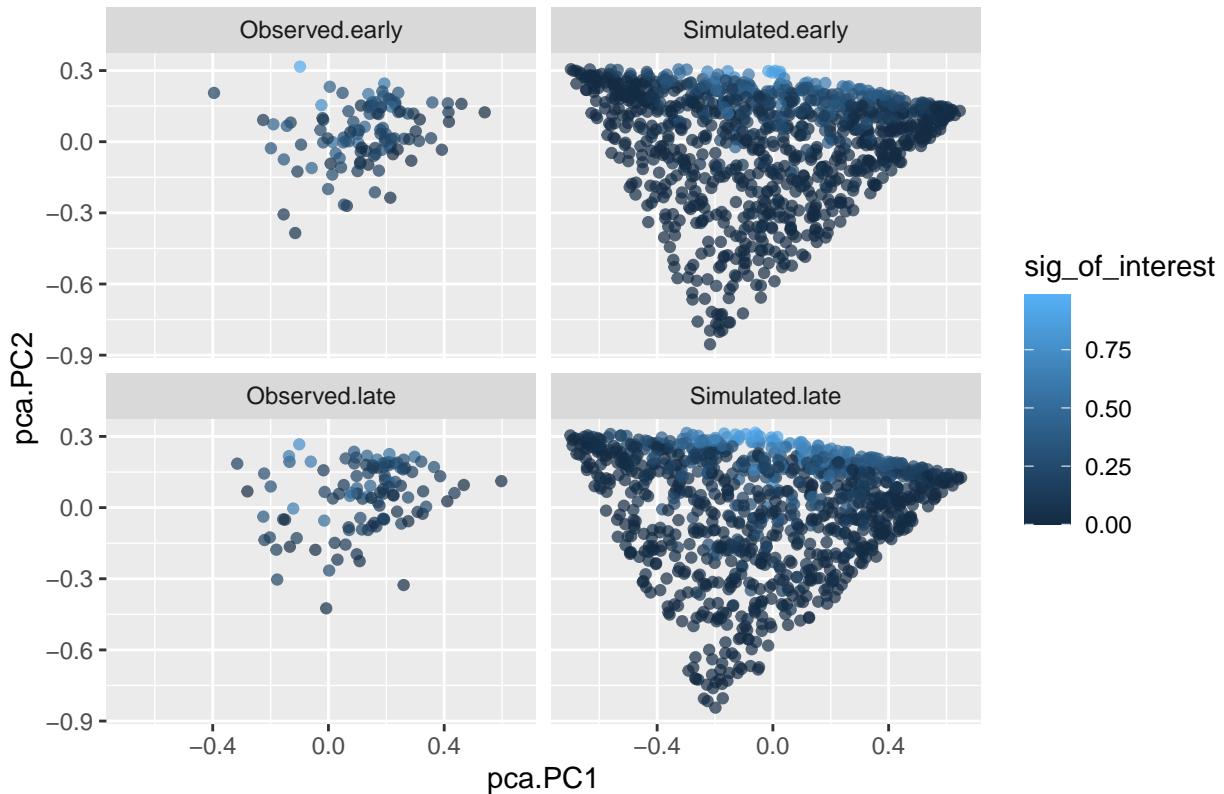
Covariance matrices



Simulation under inferred data

```
## [1] 106
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

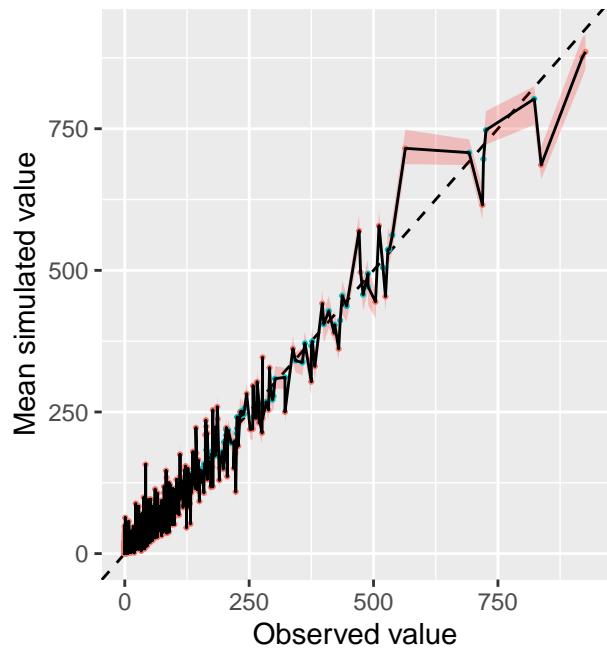
## Simulation of CNS–Medullo samples



## Ranked plot for coverage

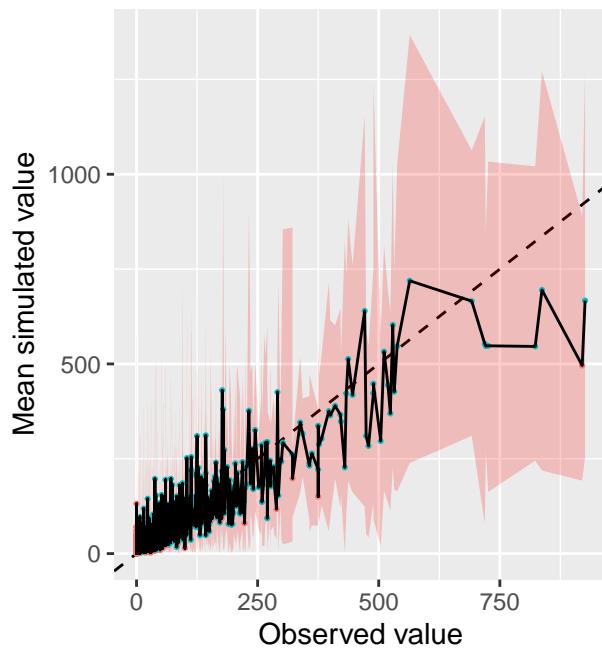
```
ct <- "CNS-Medullo"
integer_overdispersion_param_DMSL <- 1
obj_CNS_Medullo_nonexo <- give_subset_sigs_TMBobj(obj_CNS_Medullo, sigs_to_remove = nonexogenous$V1)
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_CNS_Medullo_nonexo,
print_plot = F, nreps = 20, model = "M")), function(i){
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )}[[1]],
data_object = obj_CNS_Medullo_nonexo,
loglog = F, title = 'obj_CNS_Medullo (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_CNS_Medullo_nonexo,
print_plot = F, nreps = 20, model = "DMSL", integer_overdispersion_param = integer_overdispersion_param_DMSL),
lapply(i, function(j) cbind.data.frame(sorted_value=as.vector(j),
rank_number=1:length(j)) )}[[1]],
data_object = obj_CNS_Medullo_nonexo,
loglog = F, title = 'obj_CNS_Medullo (DMSL)', ncol=2)
```

obj\_CNS\_Medullo (M)  
FALSE:812; TRUE:460



col • FALSE • TRUE

obj\_CNS\_Medullo (DMSL)  
FALSE:300; TRUE:972



col • FALSE • TRUE

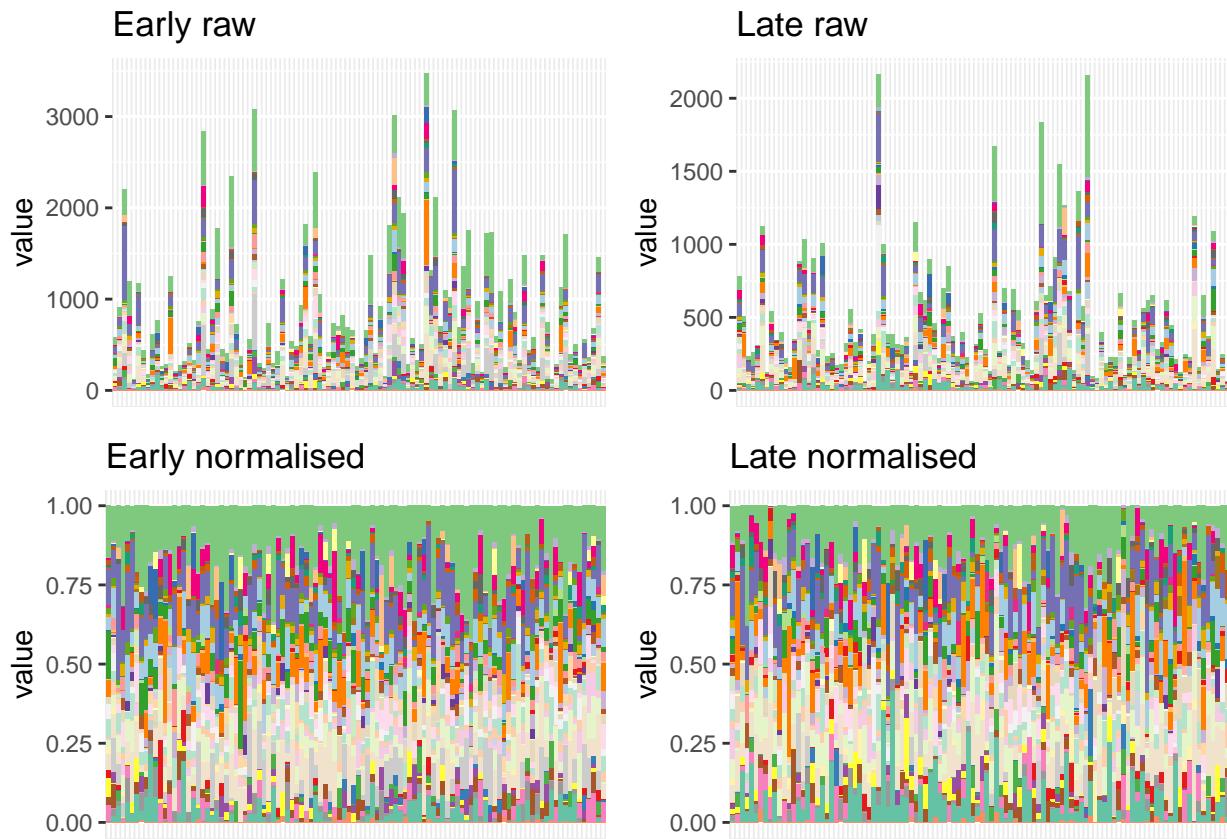
### Signatures from mutSigExtractor

The signatures from mutSigExtractor are a bit more chaotic:

```
obj_CNS_Medullo_mutSigExtractor <- load_PCAWG(ct = ct, typedata = "signaturesmutSigExtractor",
                                                path_to_data = "../..../data/")

## [1] 106
give_barplot_from_obj(obj = obj_CNS_Medullo_mutSigExtractor, legend_on = FALSE)

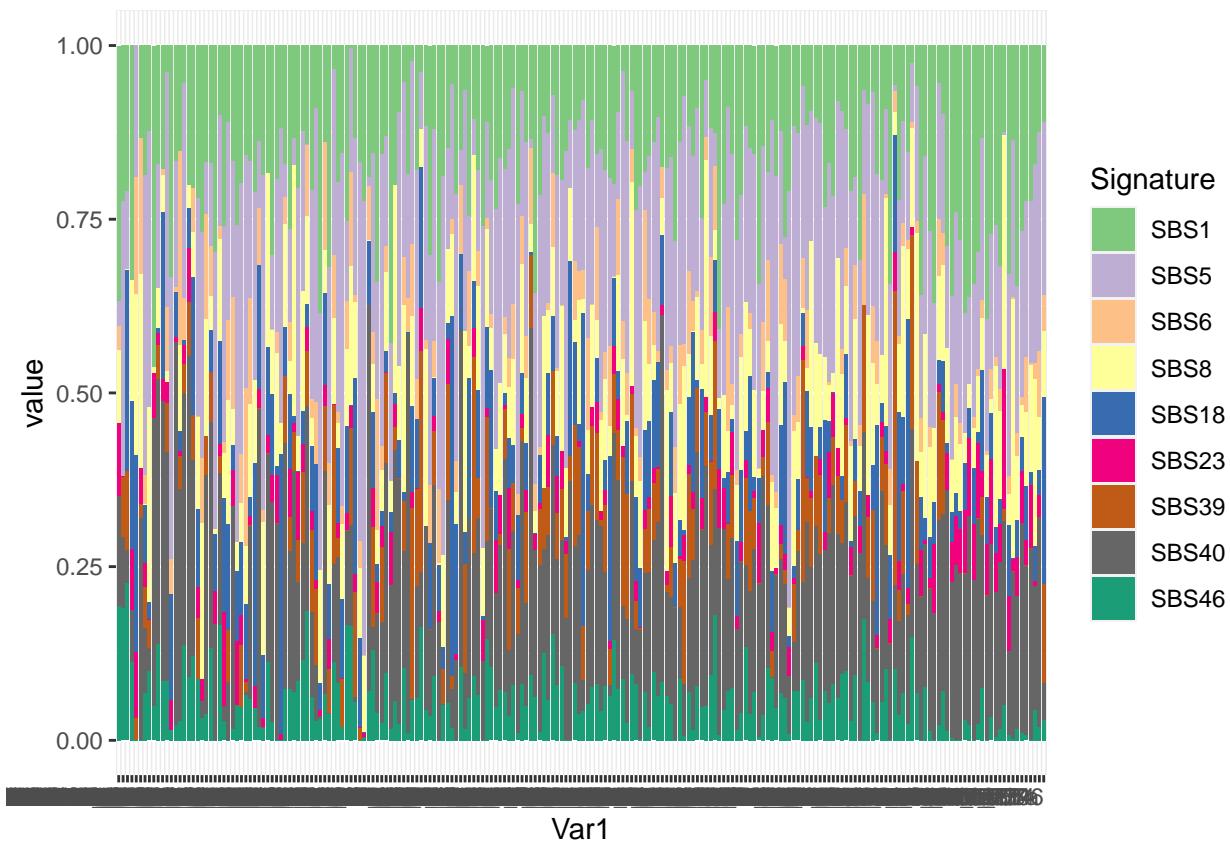
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
```



Exposures sorted by increasing number of mutations: there is no trend of signatures being associated with the number of mutations.

```
createBarplot(normalise_rw(non_duplicated_rows(obj_CNS_Medullo$Y)),
              order_labels = names(sort(rowSums(non_duplicated_rows(obj_CNS_Medullo$Y)),
                                         decreasing = F)))
```

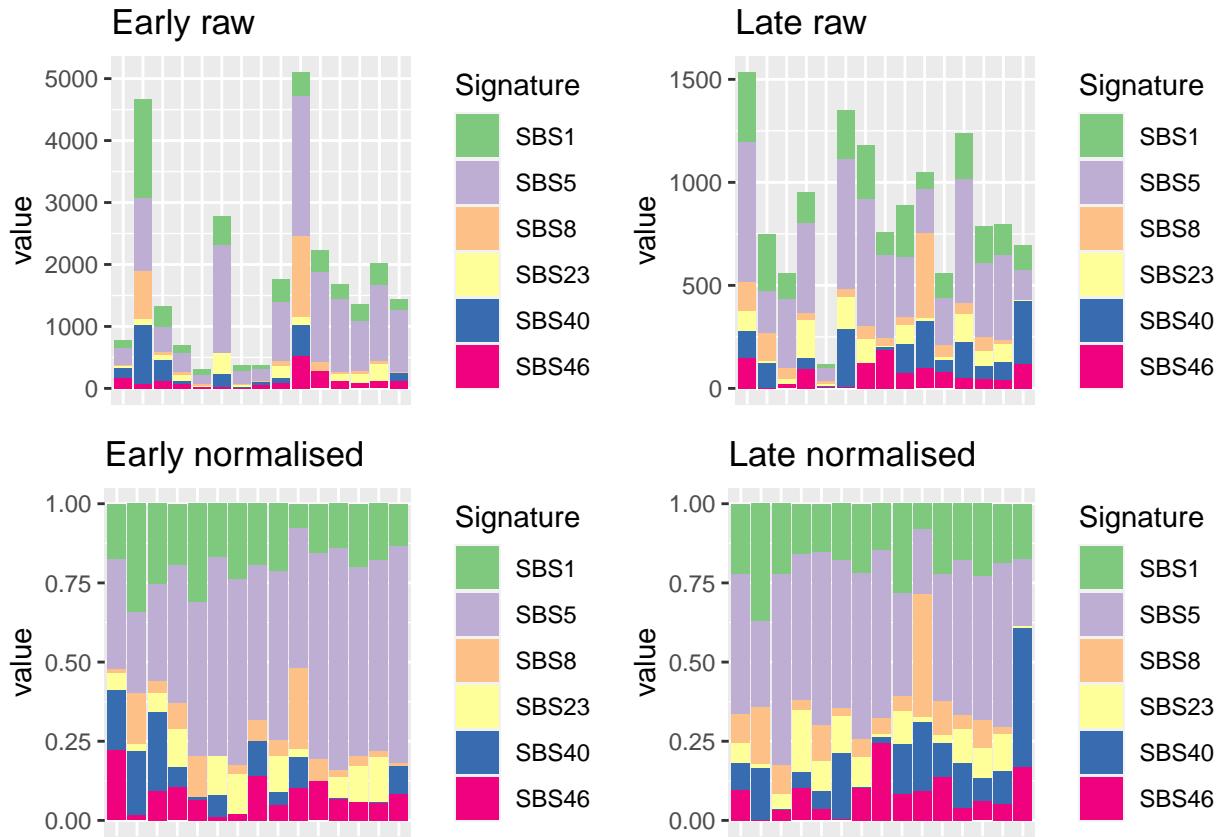
```
## Creating plot... it might take some time if the data are large. Number of samples: 212
```



## CNS-Oligo

### Barplot and general statistics

```
## [1] 15
## Creating plot... it might take some time if the data are large. Number of samples: 15
## Creating plot... it might take some time if the data are large. Number of samples: 15
## Creating plot... it might take some time if the data are large. Number of samples: 15
## Creating plot... it might take some time if the data are large. Number of samples: 15
```



The number of samples and signatures is:

```
## [1] 30 6
```

The signatures are:

```
## [1] "SBS1" "SBS5" "SBS8" "SBS23" "SBS40" "SBS46"
```

### Convergence table

We only have converged results for the multinomial with full RE, and the DM with a single lambda (diag and sparse RE). It is the same for nonexogenous signatures.

	value	L2	L1
## 1	CNS-Oligo	hessian_positivedefinite_bool	diagRE_M
## 2	CNS-Oligo	hessian_positivedefinite_bool	fullRE_M
## 3	CNS-Oligo	hessian_positivedefinite_bool	diagRE_DMDL
## 4	CNS-Oligo	hessian_nonpositivedefinite_bool	fullRE_halfDM
## 5	CNS-Oligo	hessian_nonpositivedefinite_bool	fullRE_DMDL
## 6	CNS-Oligo	hessian_positivedefinite_bool	diagRE_DMSL
## 7	CNS-Oligo	hessian_positivedefinite_bool	sparseRE_DMSL
## 8	CNS-Oligo	hessian_positivedefinite_bool	fullRE_DMSL
## 9	CNS-Oligo	hessian_nonpositivedefinite_bool	fullRE_DMSL_SBS1
## 10	CNS-Oligo	hessian_positivedefinite_bool	fullRE_M_nonexo
## 11	CNS-Oligo	hessian_positivedefinite_bool	diagRE_DMSL_nonexo
## 12	CNS-Oligo	Timeout	sparseRE_DMSL_nonexo
## 13	CNS-Oligo	hessian_positivedefinite_bool	fullRE_DMSL_nonexo

```
## 14 CNS-Oligo      hessian_positivedefinite_bool      fullRE_DMDL_nonexo
## 15 CNS-Oligo          Timeout fullRE_DMDL_sortednonexo
```

**CNS-PiloAstro**

**ColoRect-AdenoCA**

**Eso-AdenoCA**

**Head-SCC**

**Kidney-ChRCC**

**Kidney-RCC.clearcell**

**Kidney-RCC.papillary**

**Liver-HCC**

**Lung-AdenoCA**

**Lung-SCC**

**Lymph-BNHL**

**Lymph-CLL**

**Myeloid-MPN**

**Ovary-AdenoCA**

**Panc-AdenoCA**

**Panc-Endocrine**

**Prost-AdenoCA**

**Skin-Melanoma.acral**

**Skin-Melanoma.cutaneous**

**Stomach-AdenoCA**

**Thy-AdenoCA**

**Uterus-AdenoCA**

## All p-values for non-exogenous signatures

% latex table generated in R 4.0.3 by xtable 1.8-4 package % Wed May 26 18:26:30 2021

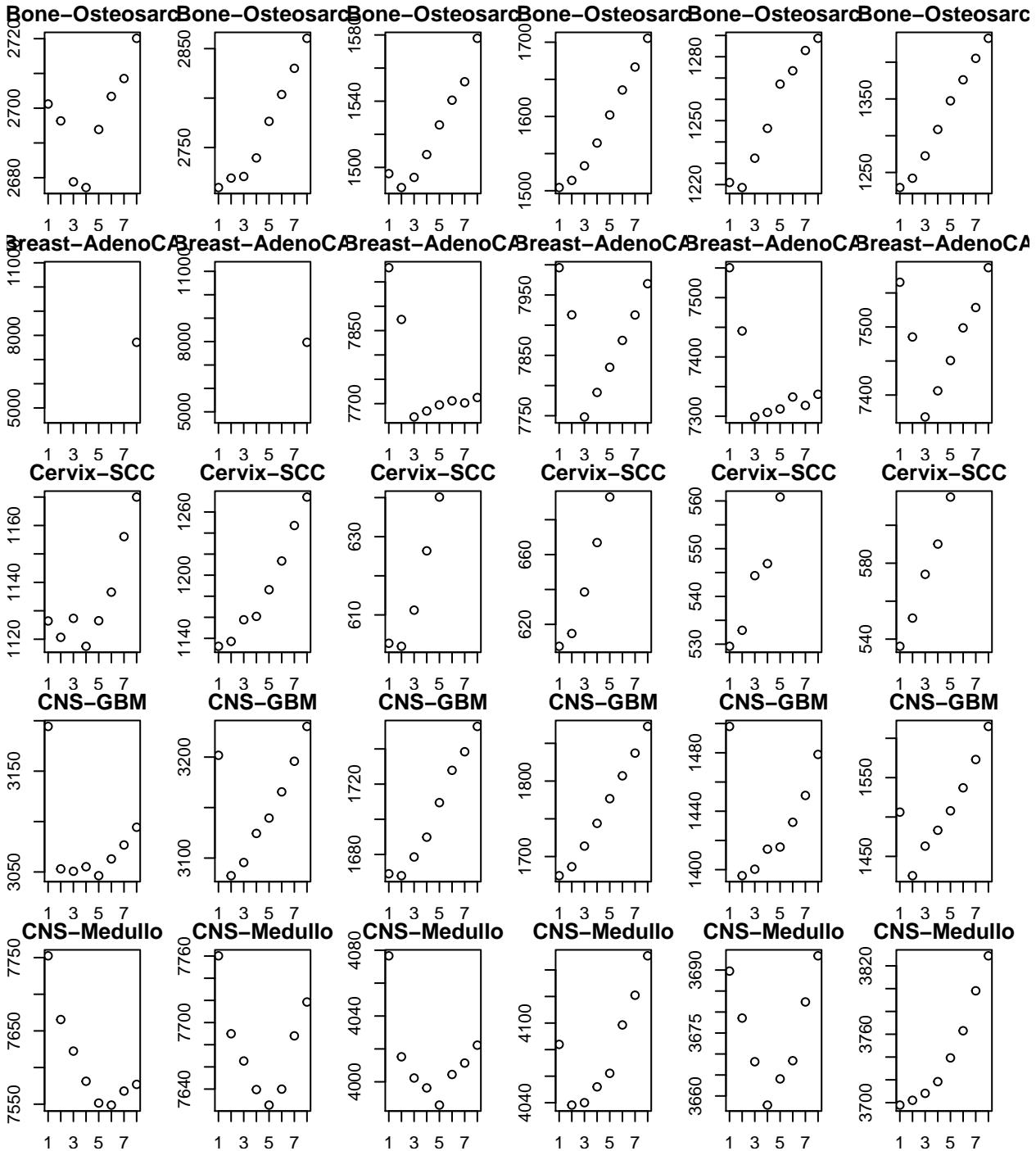
	ct	pvalue	model
1	Bone-Osteosarc	0.00	diagRE_DMSL_nonexo
2	Breast-AdenoCA	0.00	diagRE_DMSL_nonexo
3	Cervix-SCC	0.00	fullRE_DMSL_nonexo

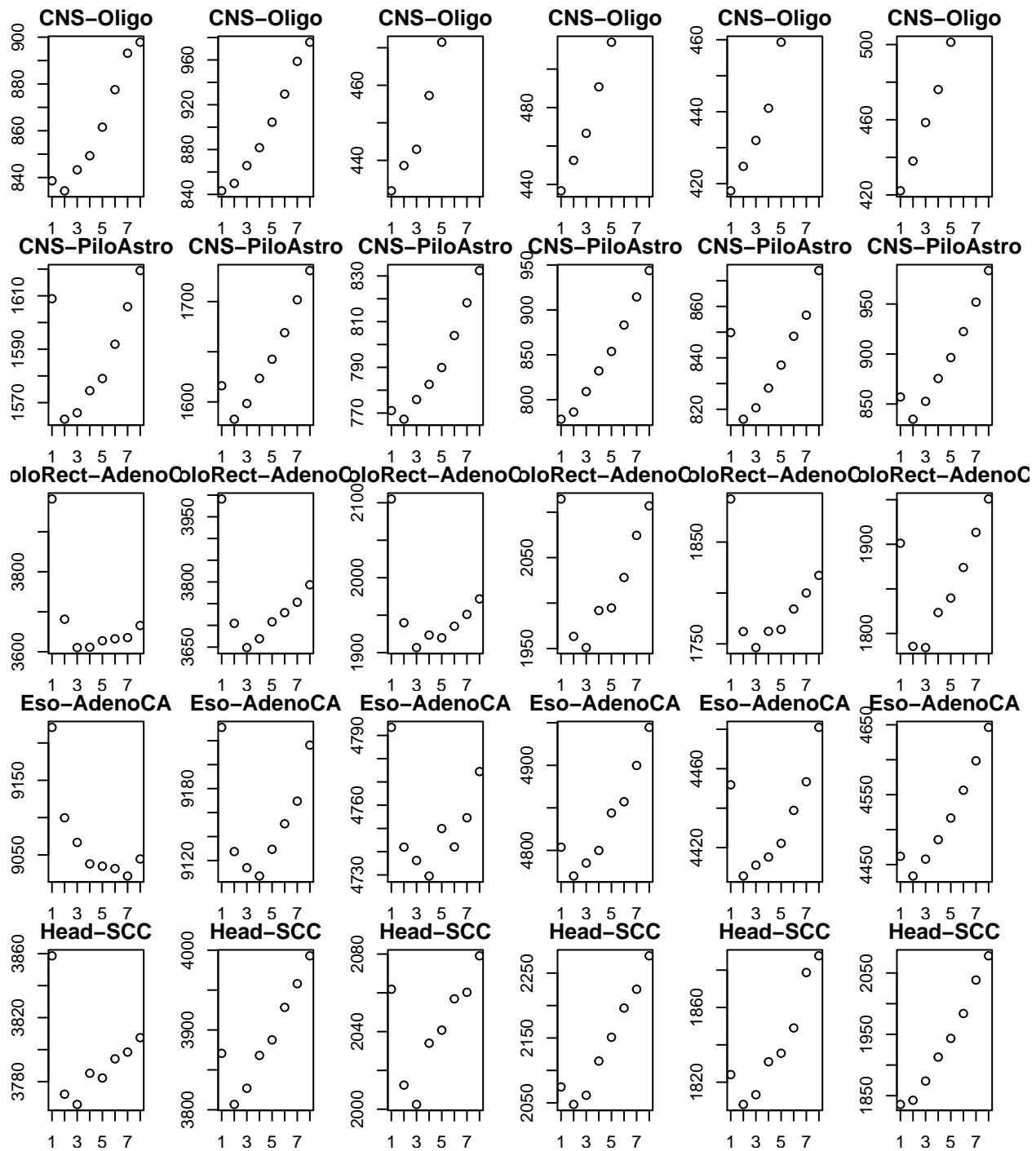
## Dirichlet-Multinomial Mixtures

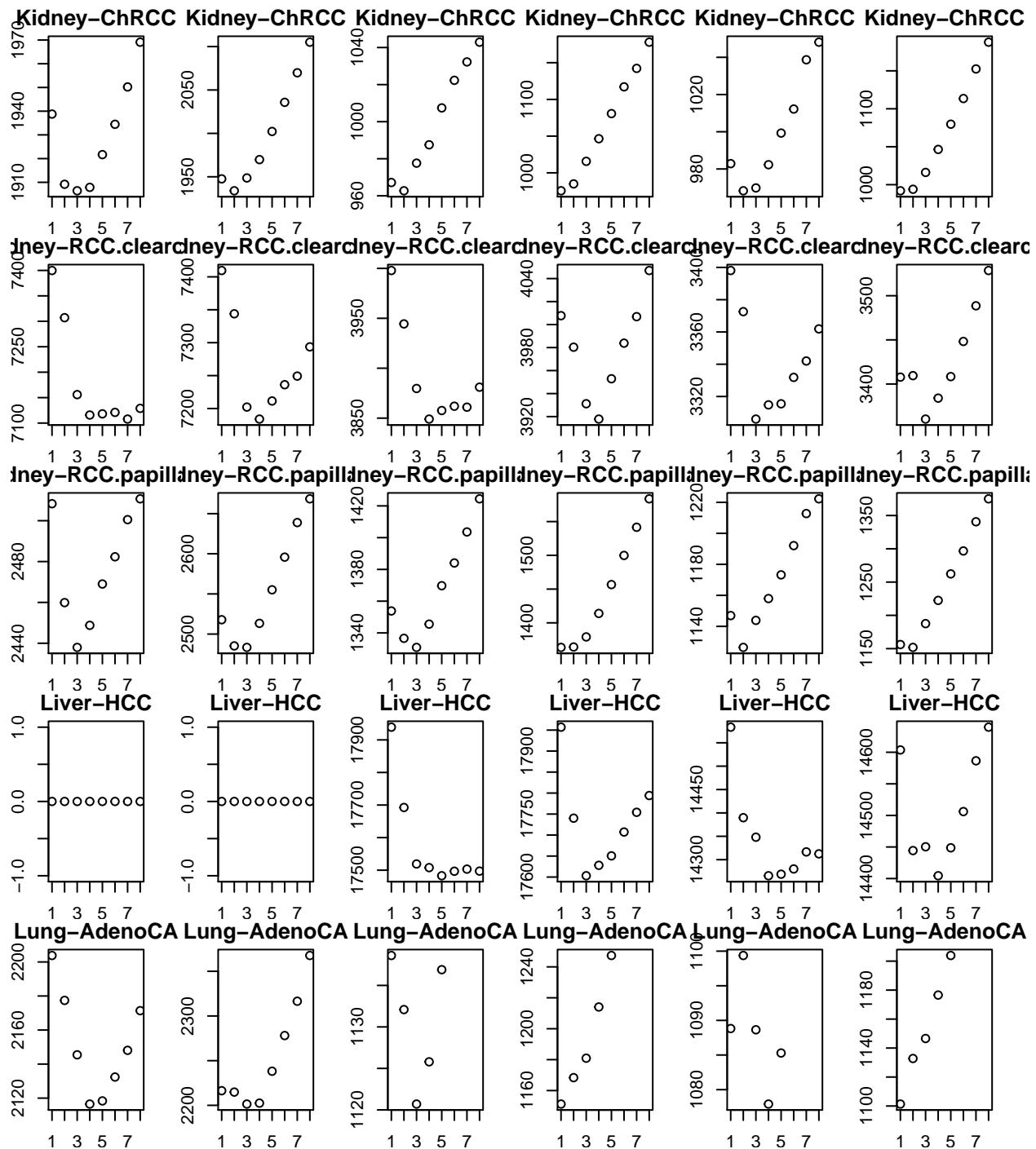
We run the software MicrobeDMMv1.0 to determine whether we are facing DMM mixtures or not.

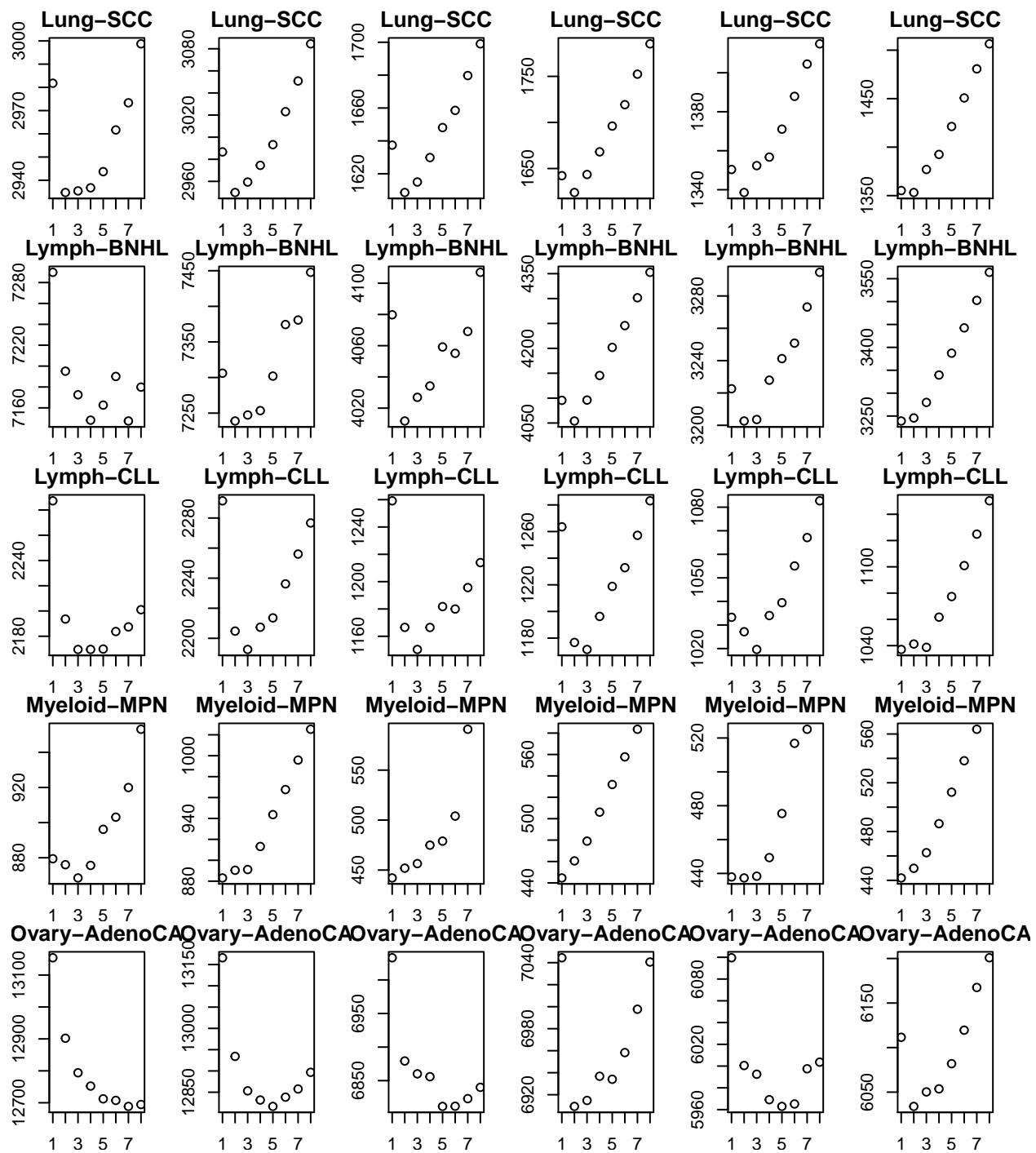
We save the files in two ways: all of the samples - early or not - together, and separately.

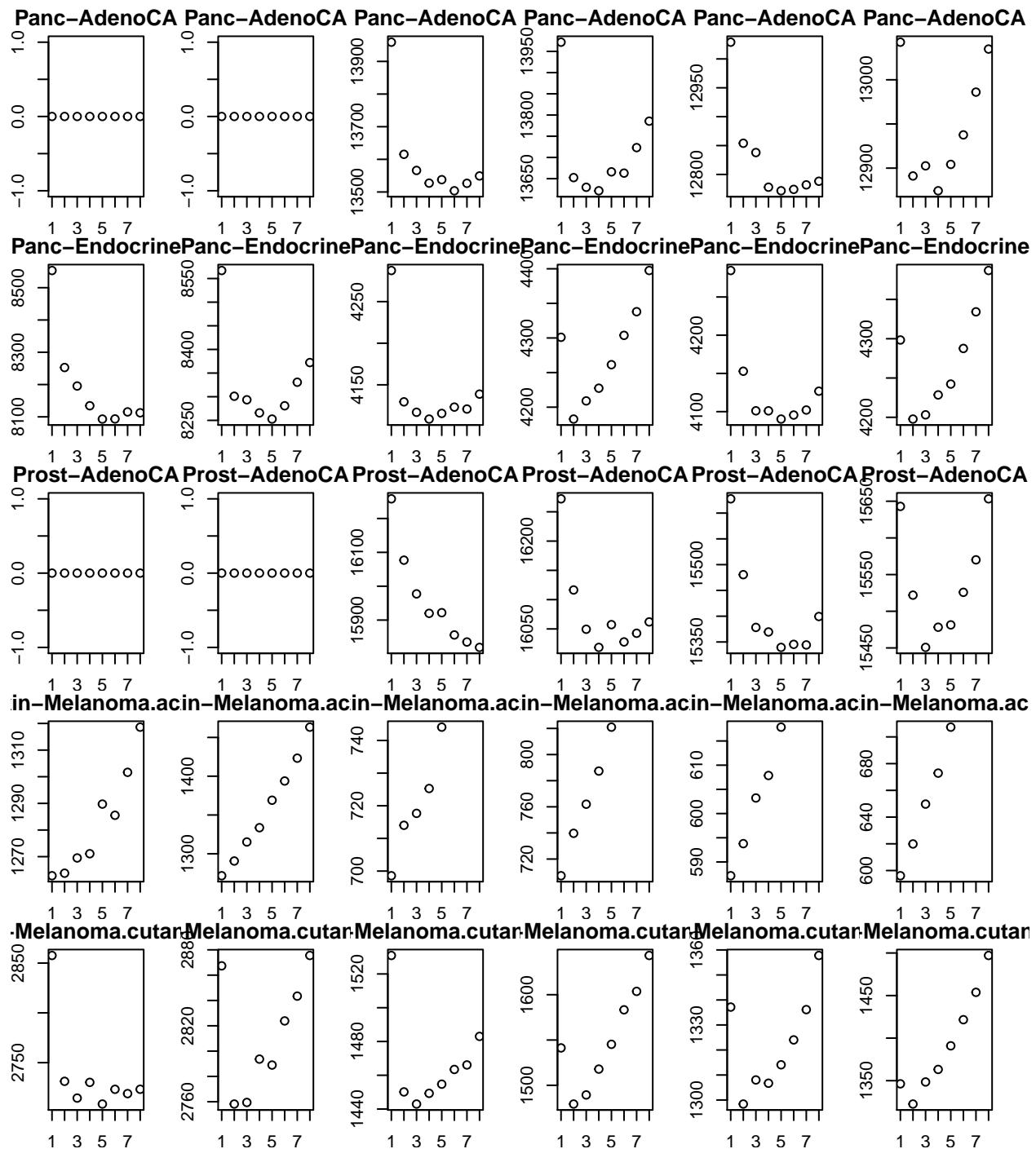
In some cases DMM says that there is an error with the input file - in this case the AIC or BIC is not plotted. If all of them are missing, all BIC and AIC are set to zero.

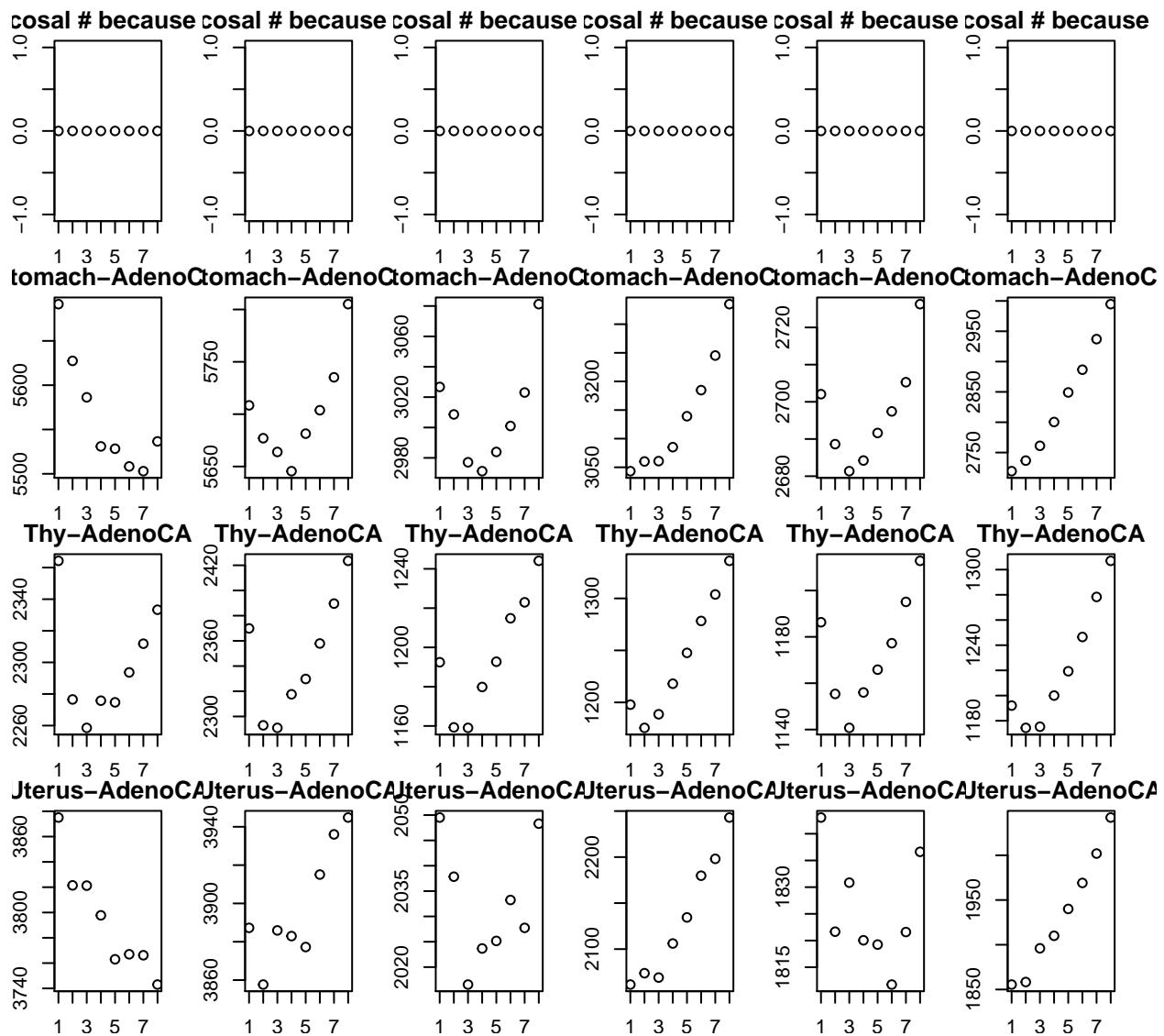












## Comparison of signature exposures with QP and mutsigextractor

```
signature_mutsigextractor_roo2 <- sapply(fles_roo[grep1('_signaturesmutSigExtractor_', fles_roo)], readRD
roo_obj2 <- roo_obj
names(signature_mutsigextractor_roo2) <- gsub("_signaturesmutSigExtractor_ROO.RDS", "", basename(names(si
names(roo_obj2) <- gsub("_signatures_ROO.RDS", "", basename(names(roo_obj2)))

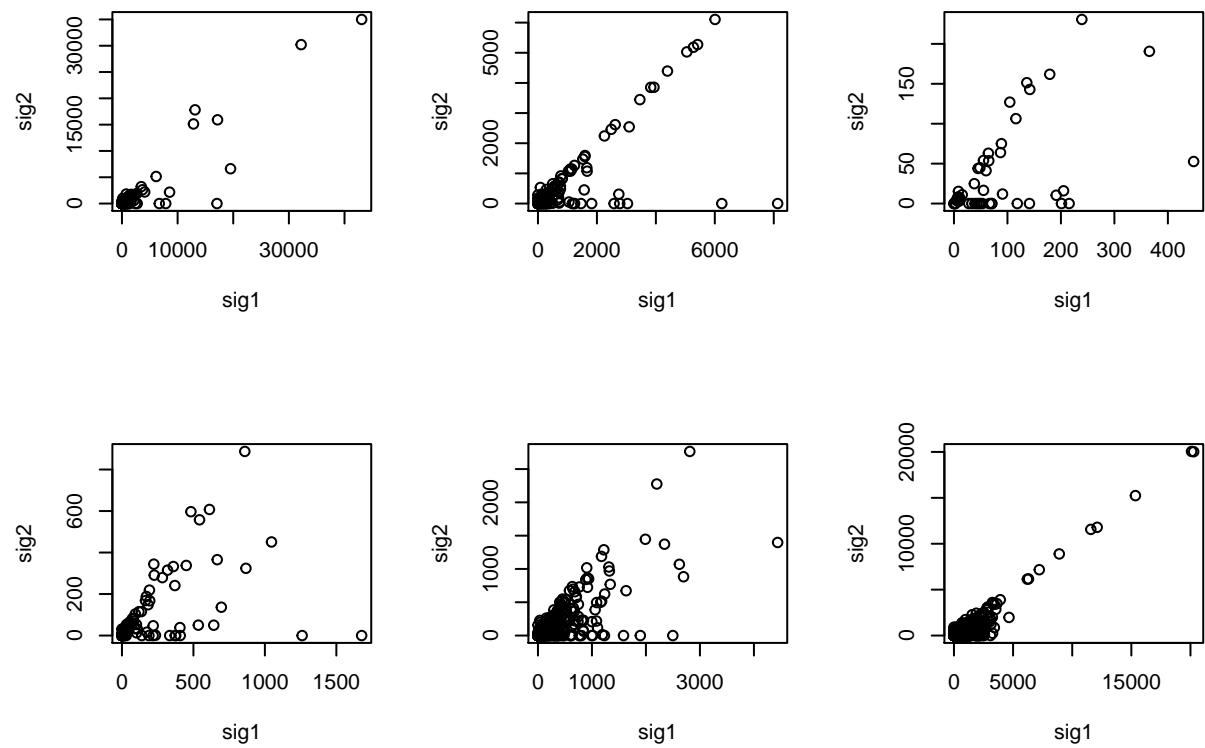
par(mfrow=c(2,3))
for(i in names(roo_obj2)){
  try({

    sig1 <- roo_obj2[[i]]
    sig2 <- signature_mutsigextractor_roo2[[i]]

    sig1 <- do.call('rbind', sig1@count_matrices_active)
    sig2 <- do.call('rbind', sig2@count_matrices_all)

    sig2 <- as.vector(sig2[,match(colnames(sig1), colnames(sig2))])
    sig1 <- as.vector(sig1)
    plot(sig1, sig2)
  })
}

## Error in do.call("rbind", sig2@count_matrices_all) :
##   trying to get slot "count_matrices_all" from an object of a basic class ("NULL") with no slots
## Error in do.call("rbind", sig2@count_matrices_all) :
##   trying to get slot "count_matrices_all" from an object of a basic class ("NULL") with no slots
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
## Warning in min(x): no non-missing arguments to min; returning Inf
## Warning in max(x): no non-missing arguments to max; returning -Inf
```



```
## Error in plot.window(...): need finite 'xlim' values
```

```
## Error in xy.coords(x, y, xlabel, ylabel, log) :  
##   'x' and 'y' lengths differ  
## Error in do.call("rbind", sig1@count_matrices_active) :  
##   trying to get slot "count_matrices_active" from an object of a basic class ("logical") with no slots  
## Error in do.call("rbind", sig2@count_matrices_all) :  
##   trying to get slot "count_matrices_all" from an object of a basic class ("NULL") with no slots
```