

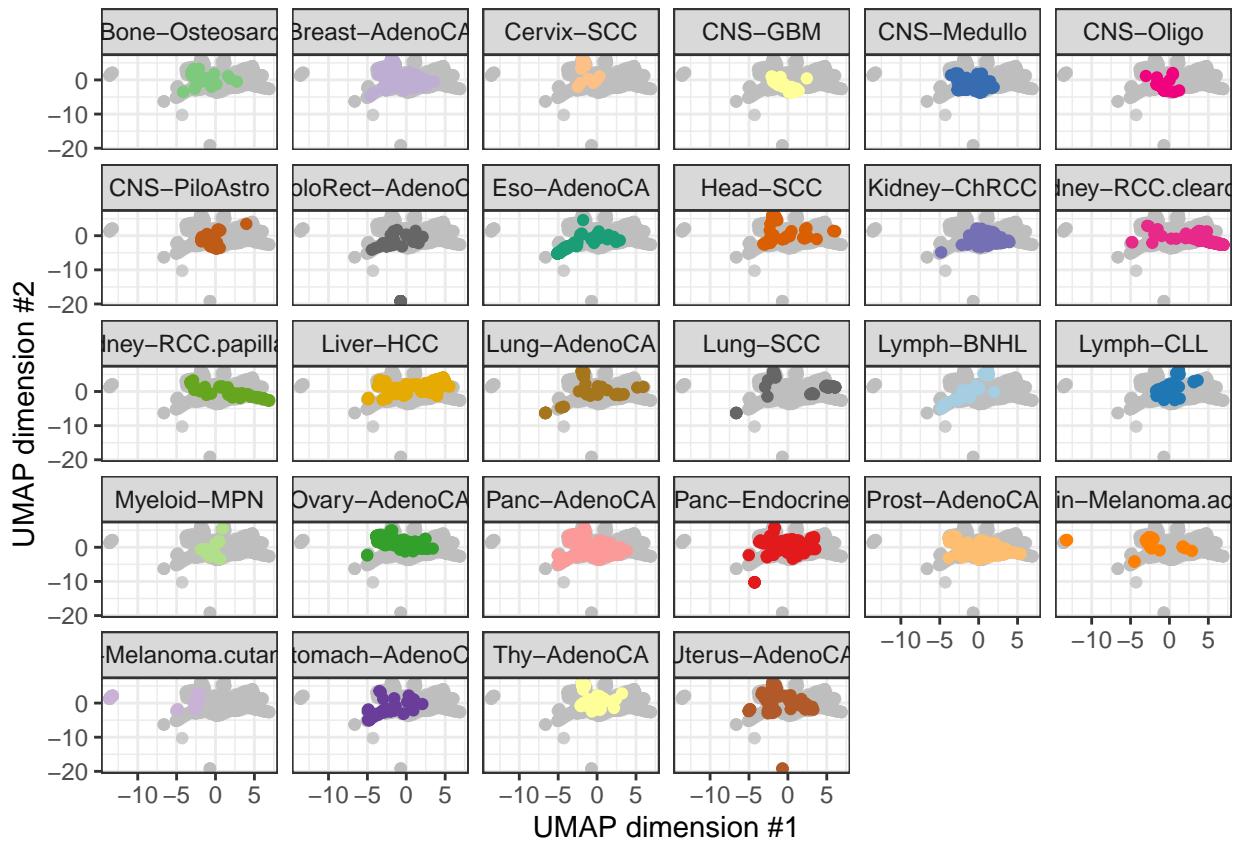
Summary of TMB runs

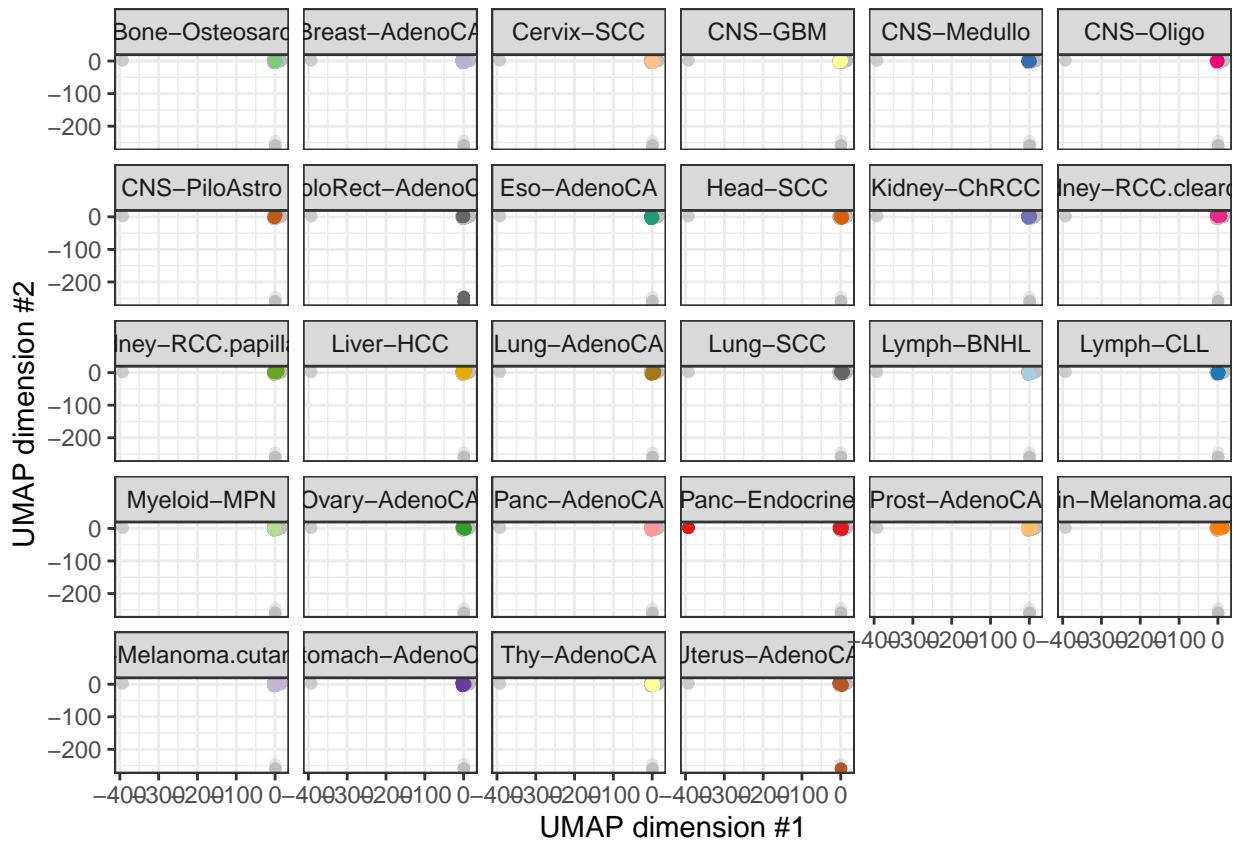
Lena Morrill

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
## Warning in .recacheSubclasses(def@class, def, env): undefined subclass
## "numericVector" of class "Mnumeric"; definition not updated
## ##
## ## 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

Information about models	7
Default order of categories for each model	7
General results of all models	8
P-values for all cancer types	9
All betas with SBS1 as baseline	11
Analysis per cancer type	15
Bone osteosarcoma	15
Barplot and general statistics	15
Convergence table	16
Re-running of fitting	17
Potentially problematic signatures	17
Betas	17
Covariance matrices	19
Simulation under inferred data	20
Ranked plot for coverage	20
Correlations of signatures	22
Signatures from mutSigExtractor	22
Breast-AdenoCA	24
Barplot and general statistics	24
Convergence table	25

Re-running of fitting	26
Potentially problematic signatures	26
Betas	27
Covariance matrices	30
Simulation under inferred data	30
Ranked plot for coverage	31
Signatures from mutSigExtractor	34
Cervix-SCC	34
Barplot and general statistics	34
Convergence table	35
Potentially problematic signatures	36
Betas	36
Covariance matrices	37
Simulation under inferred data	38
Ranked plot for coverage	39
Signatures from mutSigExtractor	40
CNS-GBM	41
Barplot and general statistics	41
Convergence table	42
Re-running of fitting	43
Potentially problematic signatures	43
Betas	43
Covariance matrices	46
Simulation under inferred data	46
Ranked plot for coverage	47
Signatures from mutSigExtractor	48
CNS-Medullo	50
Barplot and general statistics	50
Convergence table	51
Potentially problematic signatures	52
Betas	52
Covariance matrices	55
Simulation under inferred data	55
Ranked plot for coverage	56
Signatures from mutSigExtractor	57
CNS-Oligo	59
Barplot and general statistics	59
Convergence table	60
Betas	61
Covariance matrices	63
Simulation under inferred data	63
Ranked plot for coverage	64
Signatures from mutSigExtractor	65
CNS-PiloAstro	67
CNS-PiloAstro	67
Barplot and general statistics	67
Convergence table	67
Re-running of fitting	68
Potentially problematic signatures	68
Betas	68
Covariance matrices	71

Simulation under inferred data	71
Ranked plot for coverage	72
Signatures from mutSigExtractor	73
ColoRect-AdenoCA	75
ColoRect-AdenoCA	75
Barplot and general statistics	75
Convergence table	76
Re-running of fitting	77
Potentially problematic signatures	77
Betas	78
Covariance matrices	80
Simulation under inferred data	80
Ranked plot for coverage	81
Signatures from mutSigExtractor	82
Eso-AdenoCA	84
Barplot and general statistics	84
Convergence table	84
Re-running of fitting	85
Potentially problematic signatures	85
Betas	85
Covariance matrices	88
Simulation under inferred data	88
Ranked plot for coverage	89
Signatures from mutSigExtractor	90
Head-SCC	92
Barplot and general statistics	92
Convergence table	92
Re-running of fitting	93
Potentially problematic signatures	93
Betas	93
Covariance matrices	96
Simulation under inferred data	96
Ranked plot for coverage	97
Signatures from mutSigExtractor	98
Kidney-ChRCC	100
Barplot and general statistics	100
Convergence table	100
Re-running of fitting	101
Potentially problematic signatures	101
Betas	101
Covariance matrices	104
Simulation under inferred data	104
Ranked plot for coverage	105
Signatures from mutSigExtractor	106
Kidney-RCC.clearcell	108
Kidney-RCC.papillary	108
Liver-HCC	108
Lung-AdenoCA	108
Lung-SCC	108
Lymph-BNHL	108
Lymph-CLL	108

Barplot and general statistics	108
Convergence table	109
Potentially problematic signatures	109
Betas	109
Covariance matrices	112
Simulation under inferred data	112
Ranked plot for coverage	113
Signatures from mutSigExtractor	118
Myeloid-MPN	119
Ovary-AdenoCA	119
Panc-AdenoCA	119
Panc-Endocrine	119
Prost-AdenoCA	119
Skin-Melanoma.acral	119
Skin-Melanoma.cutaneous	119
Stomach-AdenoCA	119
Thy-AdenoCA	119
Uterus-AdenoCA	119
All p-values for non-exogenous signatures	119
Dirichlet-Multinomial Mixtures	120
Comparison of signature exposures with QP and mutsigextractor	126

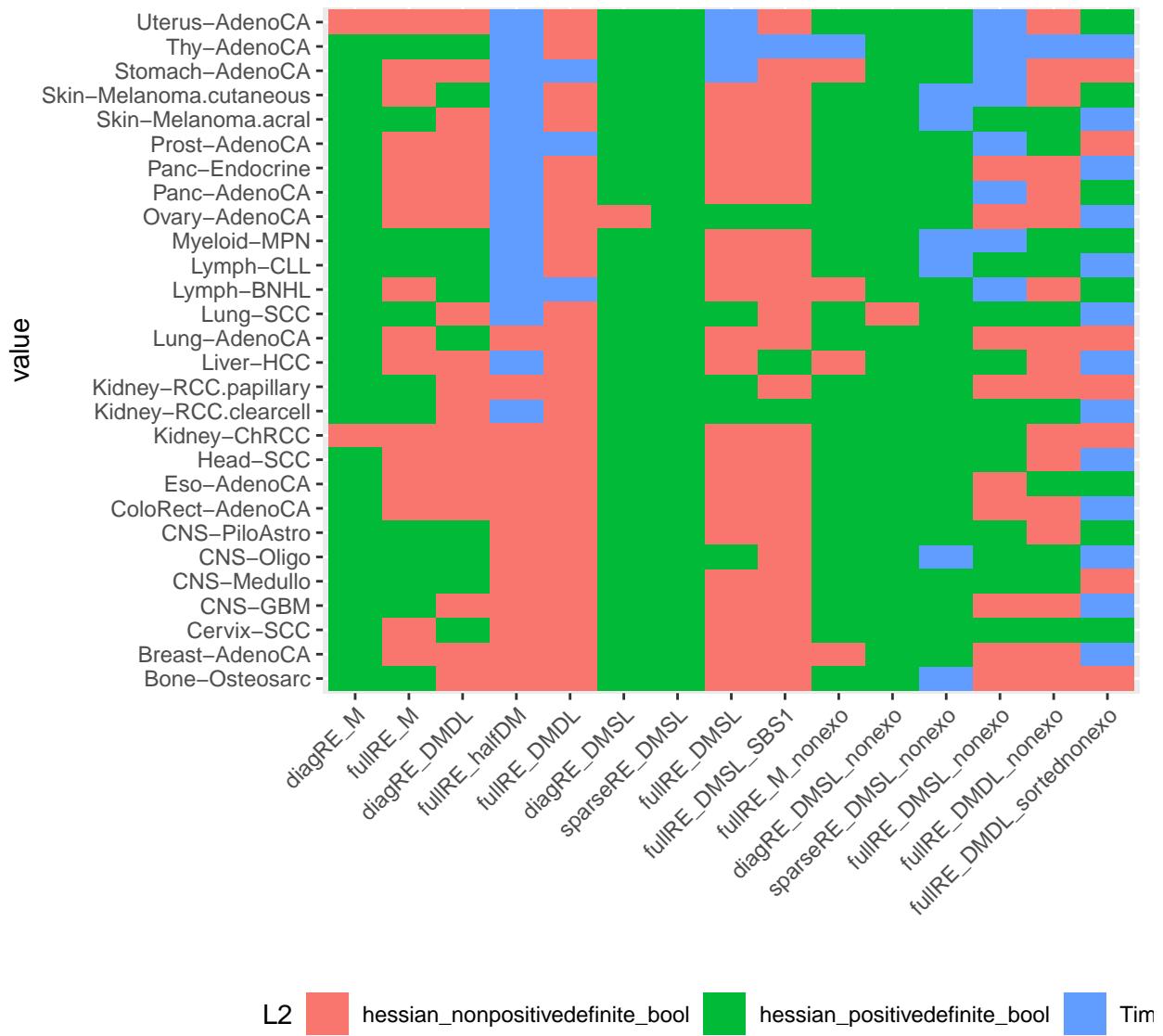
Information about models

Default order of categories for each model

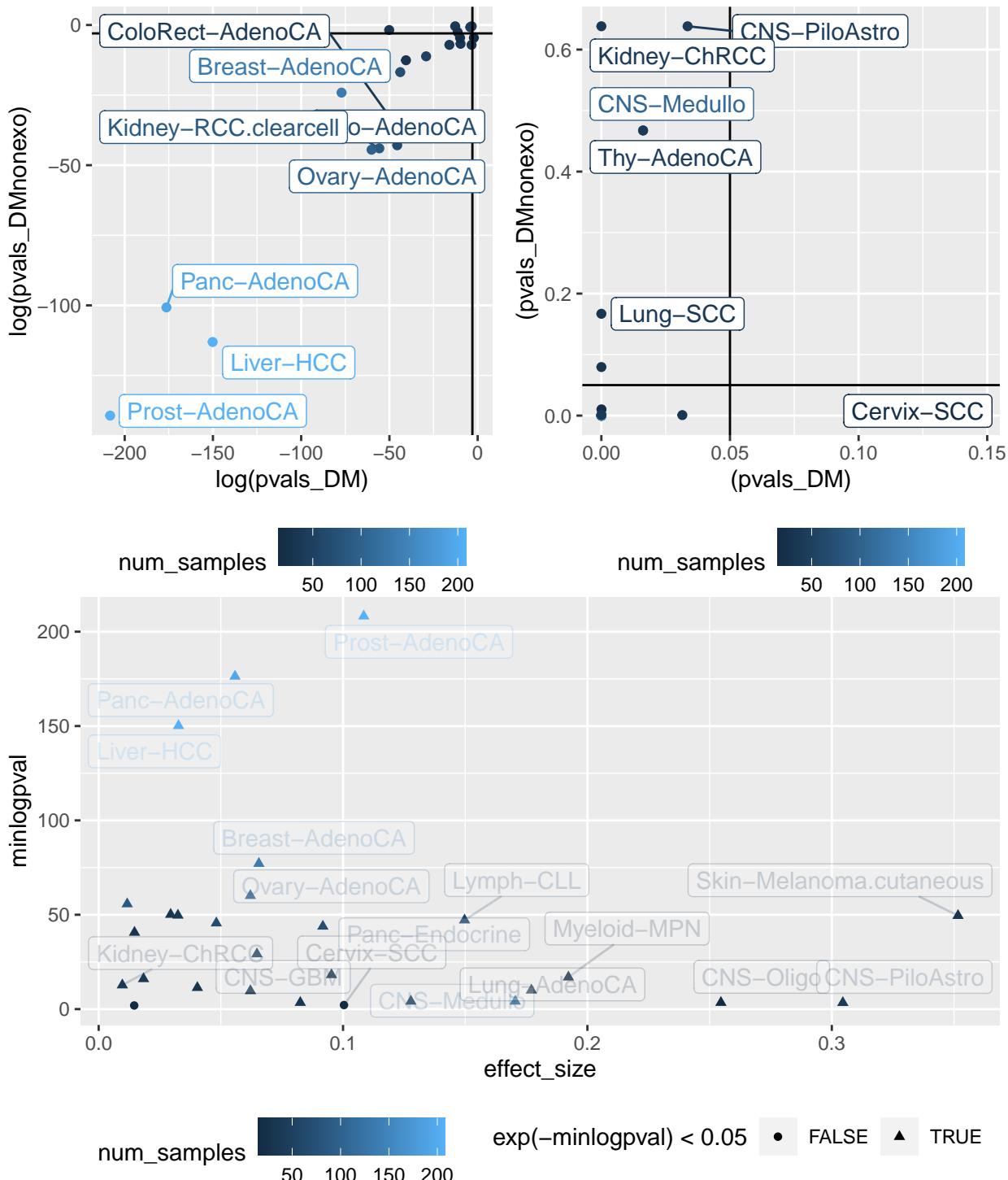
Name model	Extension	Sorted	File in which they were created
fullREDMsinglelambda	fullRE_DMSL_	Not sorted	run_TMB_PCAWG.R
fullREDMsinglelambda2	fullRE_DMSL2_	Sorted	run_TMB_PCAWG.R
diagREDMsinglelambda	diagRE_DMSL_	Unknown	run_TMB_PCAWG.R
fullRE_M	fullRE_M_	Sorted in previous version of wrapper_run_TMB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

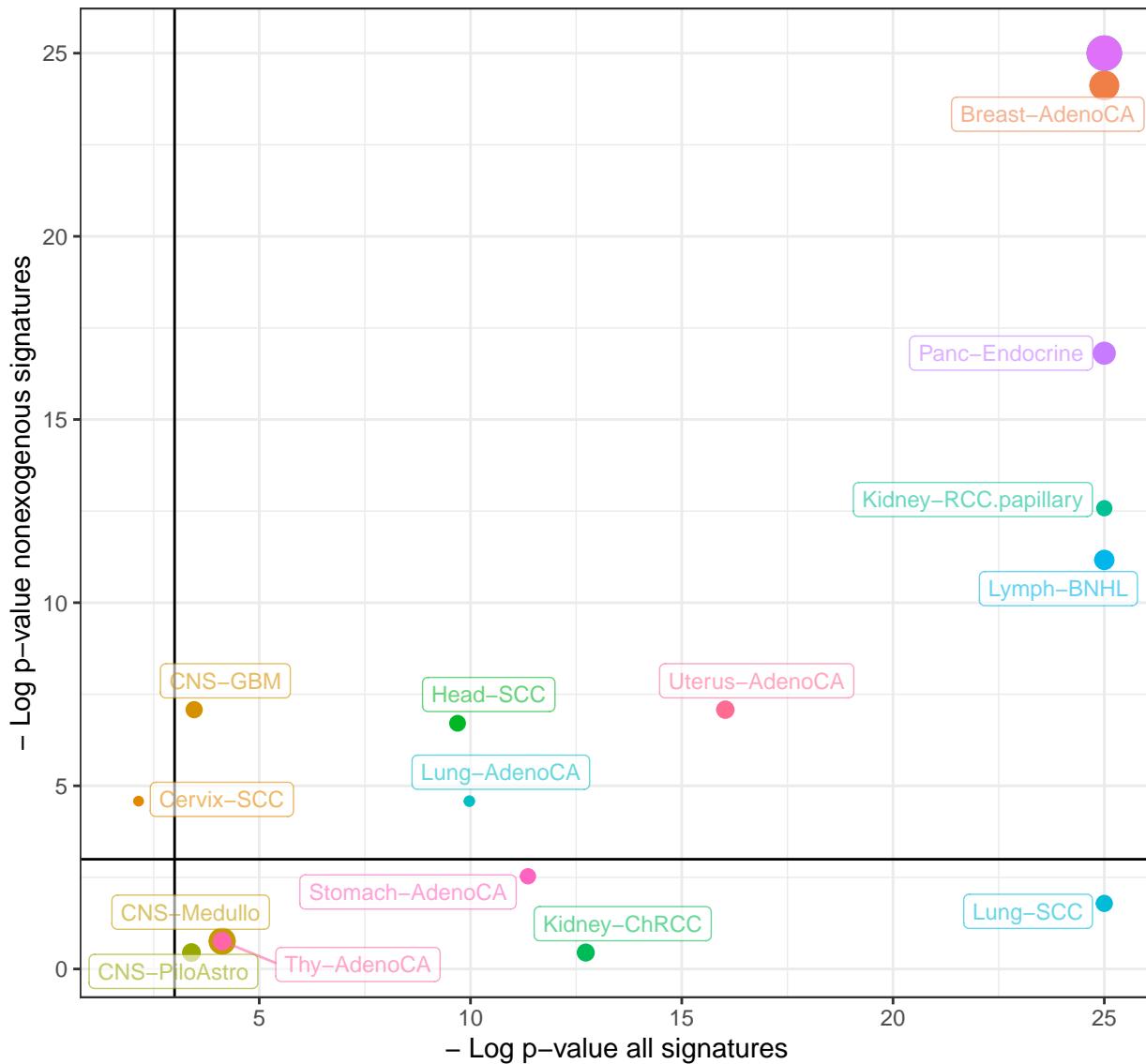
General results of all models

Check the results of all of the models



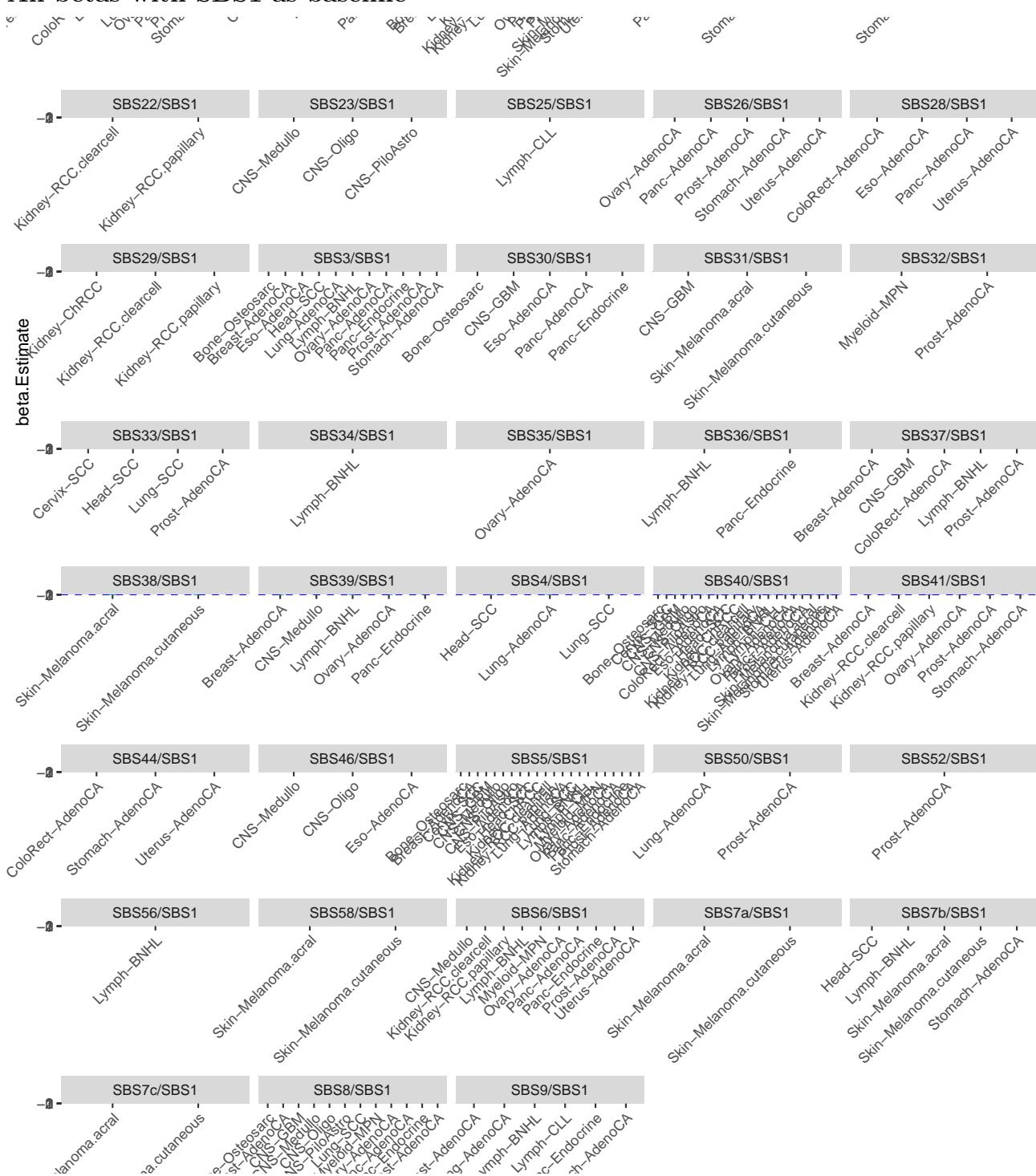
P-values for all cancer types

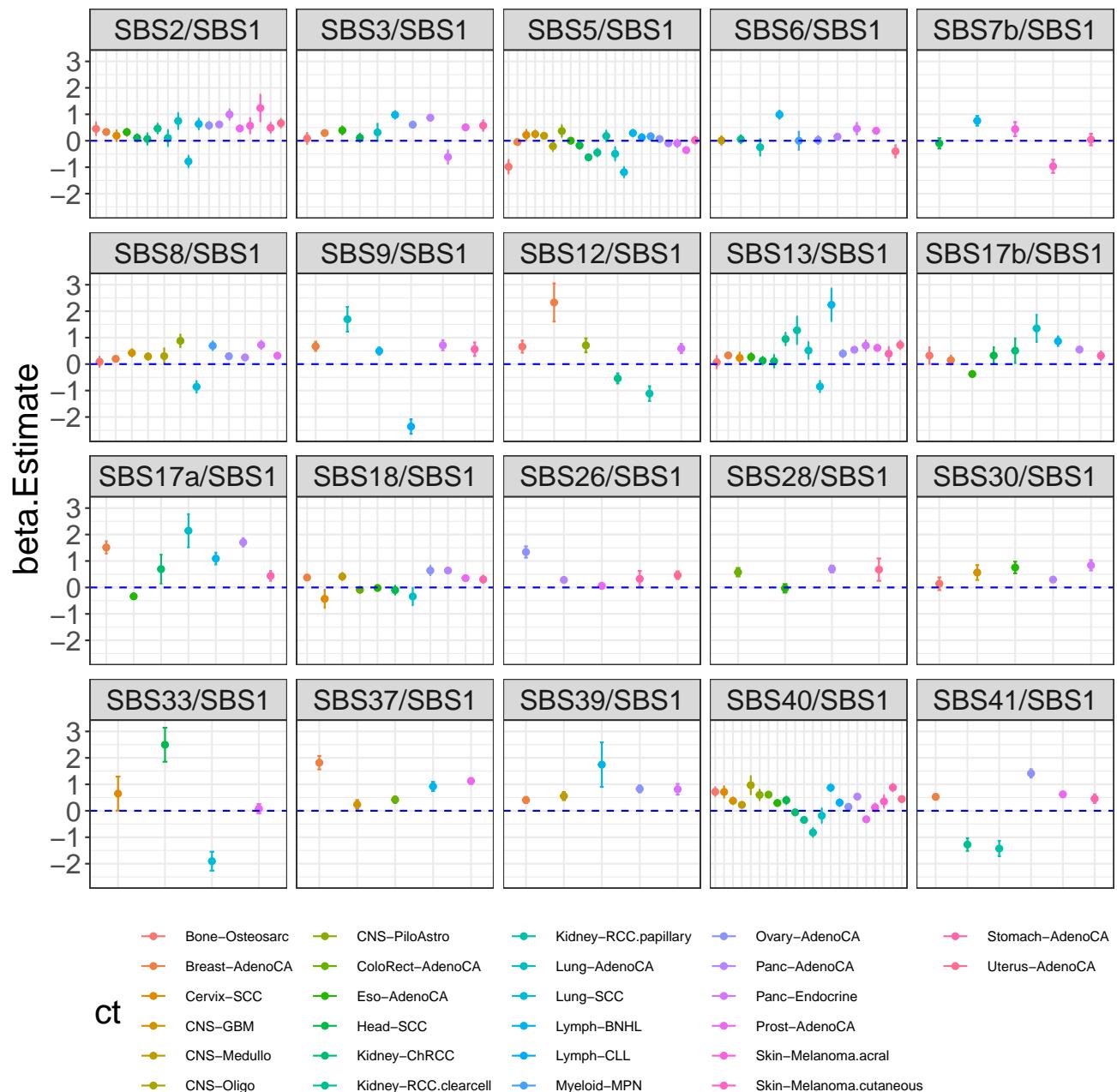


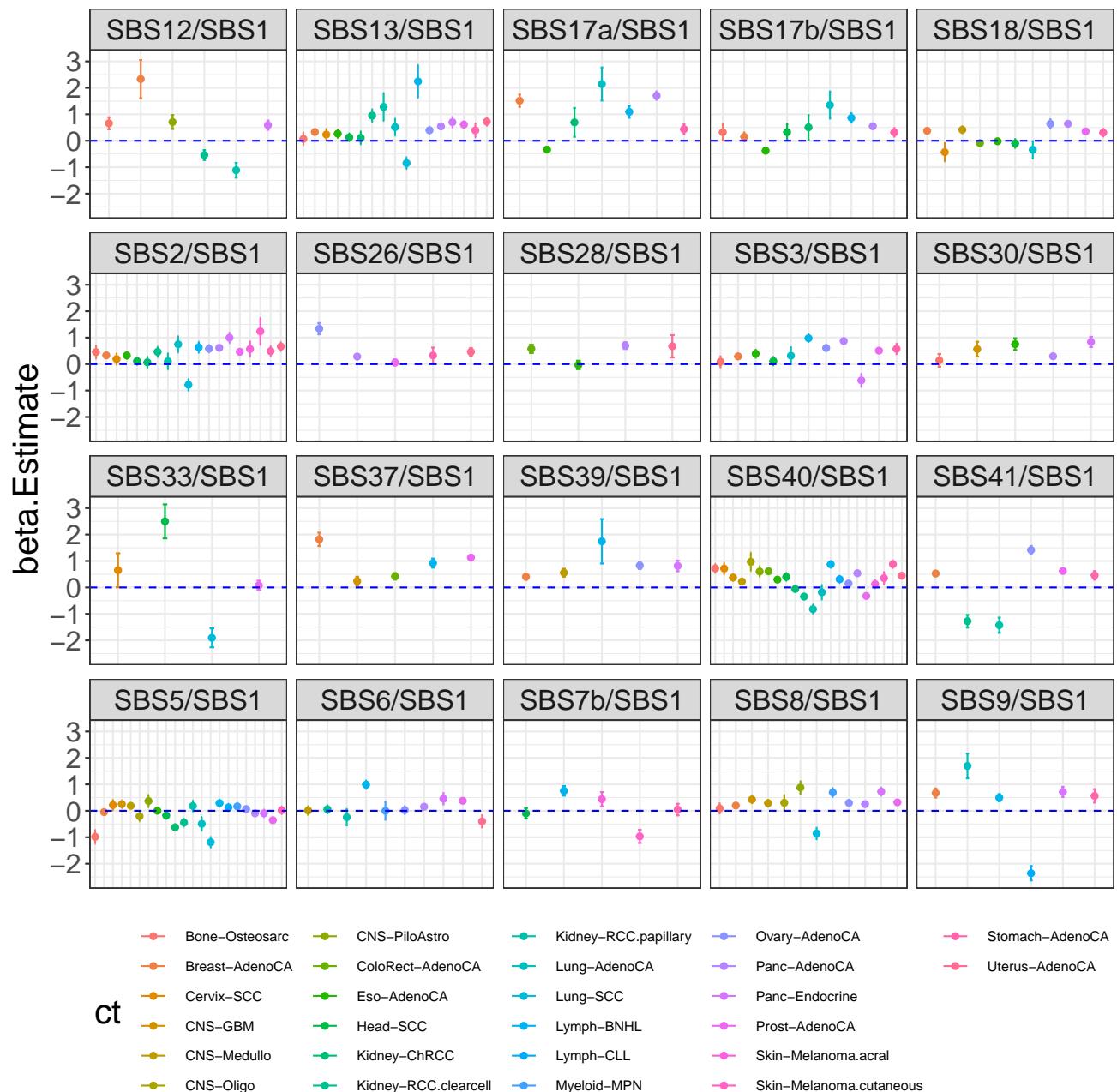


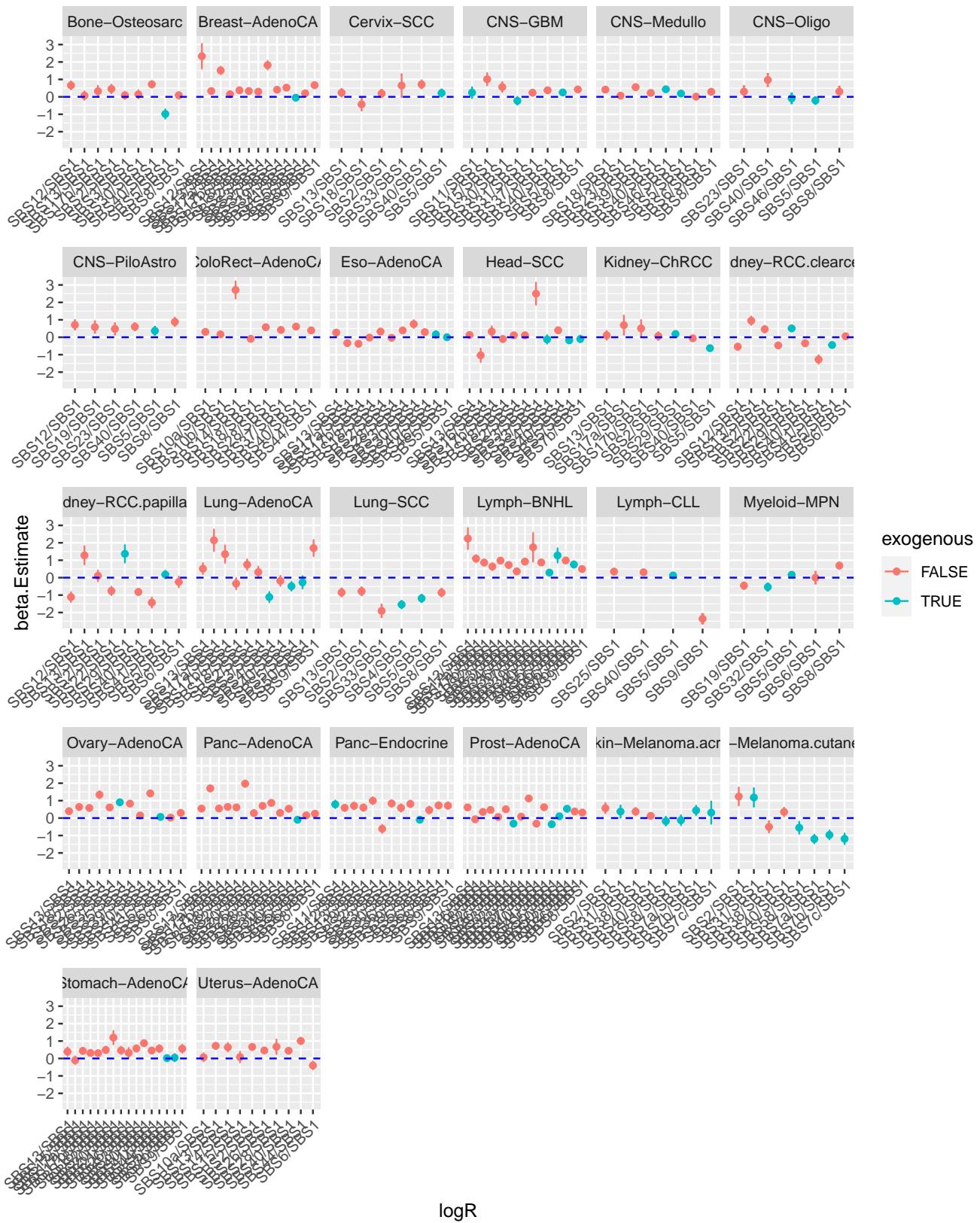
	Bone–Osteosarc	ColoRect–AdenoCA	Lung–AdenoCA	Panc–Endocrine
	Breast–AdenoCA	Eso–AdenoCA	Lung–SCC	Prost–AdenoCA
	Cervix–SCC	Head–SCC	Lymph–BNHL	Skin–Melanoma.acral
ct	CNS–GBM	Kidney–ChRCC	Lymph–CLL	Skin–Melanoma.cutaneous
	CNS–Medullo	Kidney–RCC.clearcell	Myeloid–MPN	Stomach–AdenoCA
	CNS–Oligo	Kidney–RCC.papillary	Ovary–AdenoCA	Thy–AdenoCA
	CNS–PiloAstro	Liver–HCC	Panc–AdenoCA	Uterus–AdenoCA

All betas with SBS1 as baseline



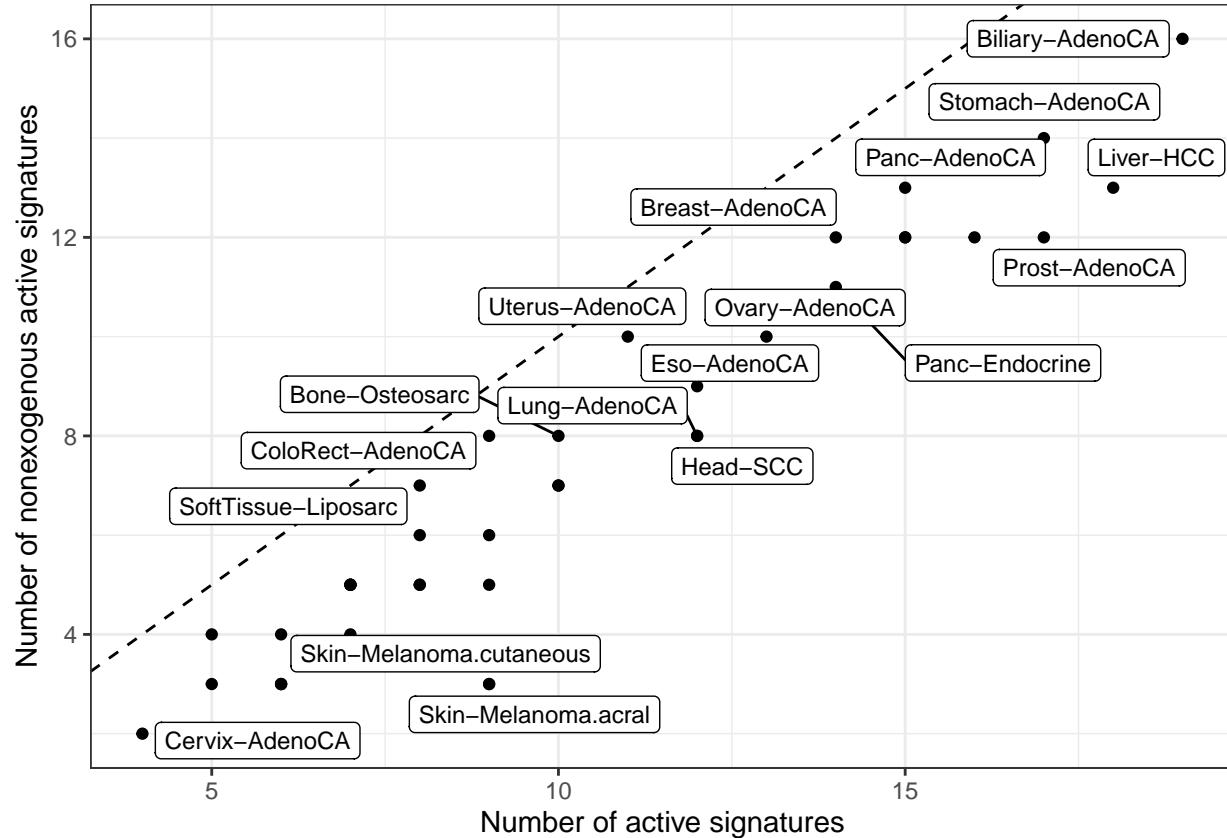






How many signatures so we have in total and how many nonexogenous ones?

```
## Error in slot(i, "count_matrices_active") :
##   cannot get a slot ("count_matrices_active") from an object of type "logical"
## Error in signature_roo_active[[j]][[1]][, !(colnames(signature_roo_active[[j]][[1]])) %in% :
##   incorrect number of dimensions
## Error in signature_roo_active[[j]][[1]][, !(colnames(signature_roo_active[[j]][[1]])) %in% :
##   incorrect number of dimensions
```

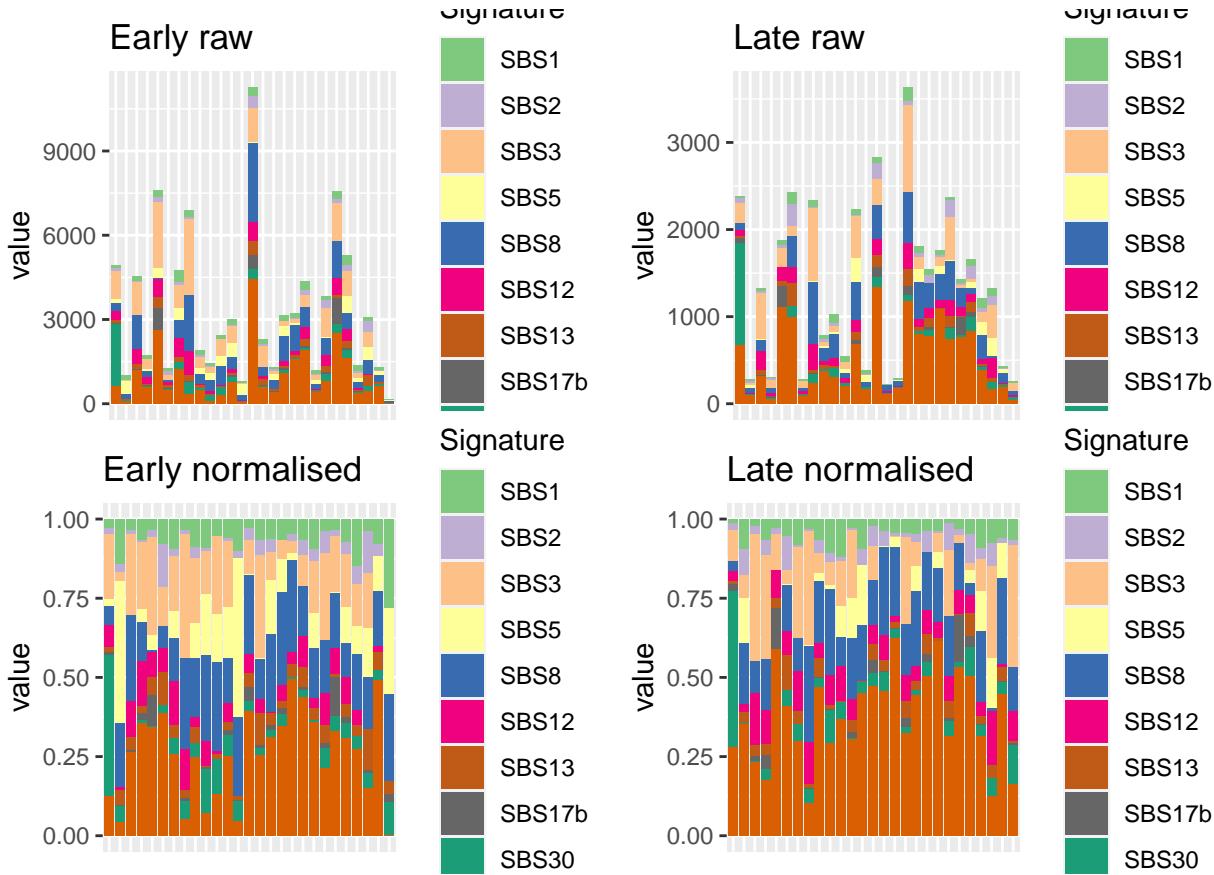


Analysis per cancer type

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%

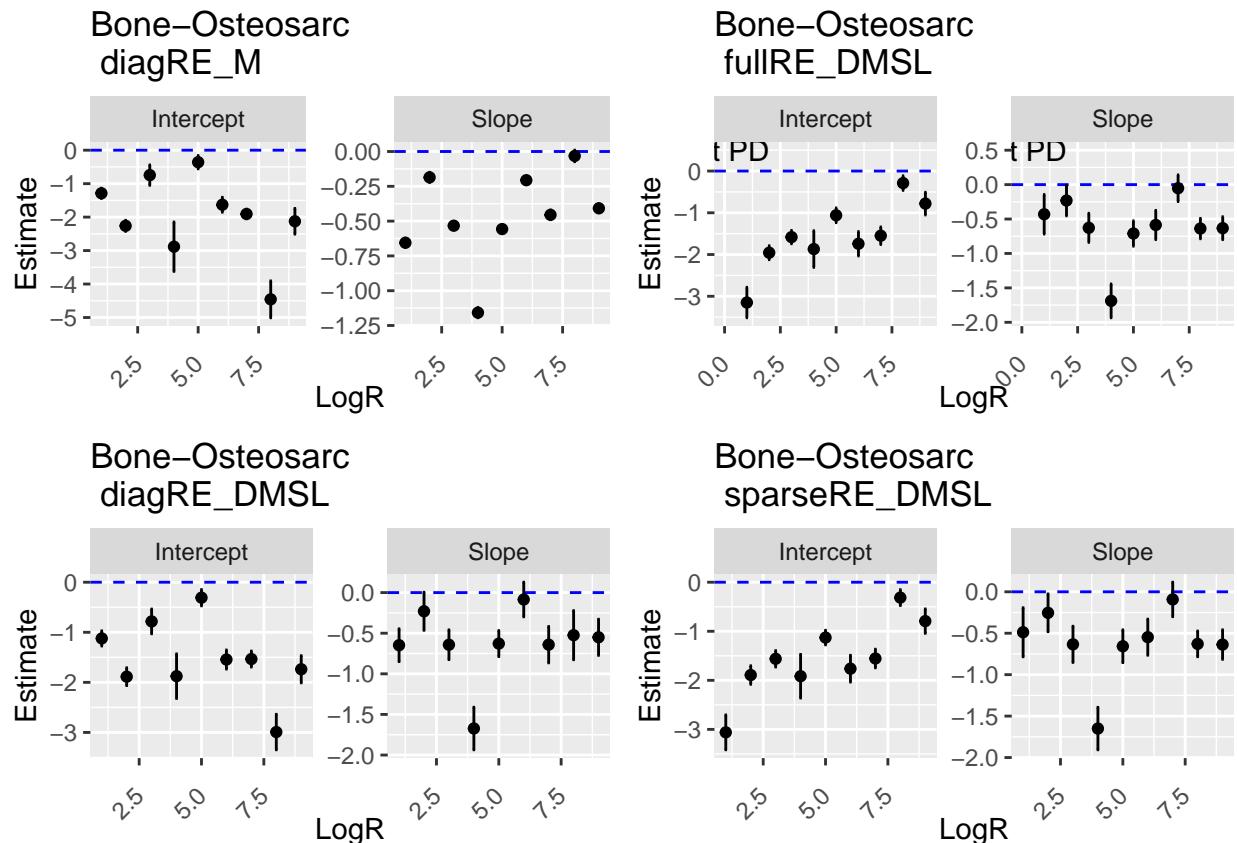
Betas

```

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)

```

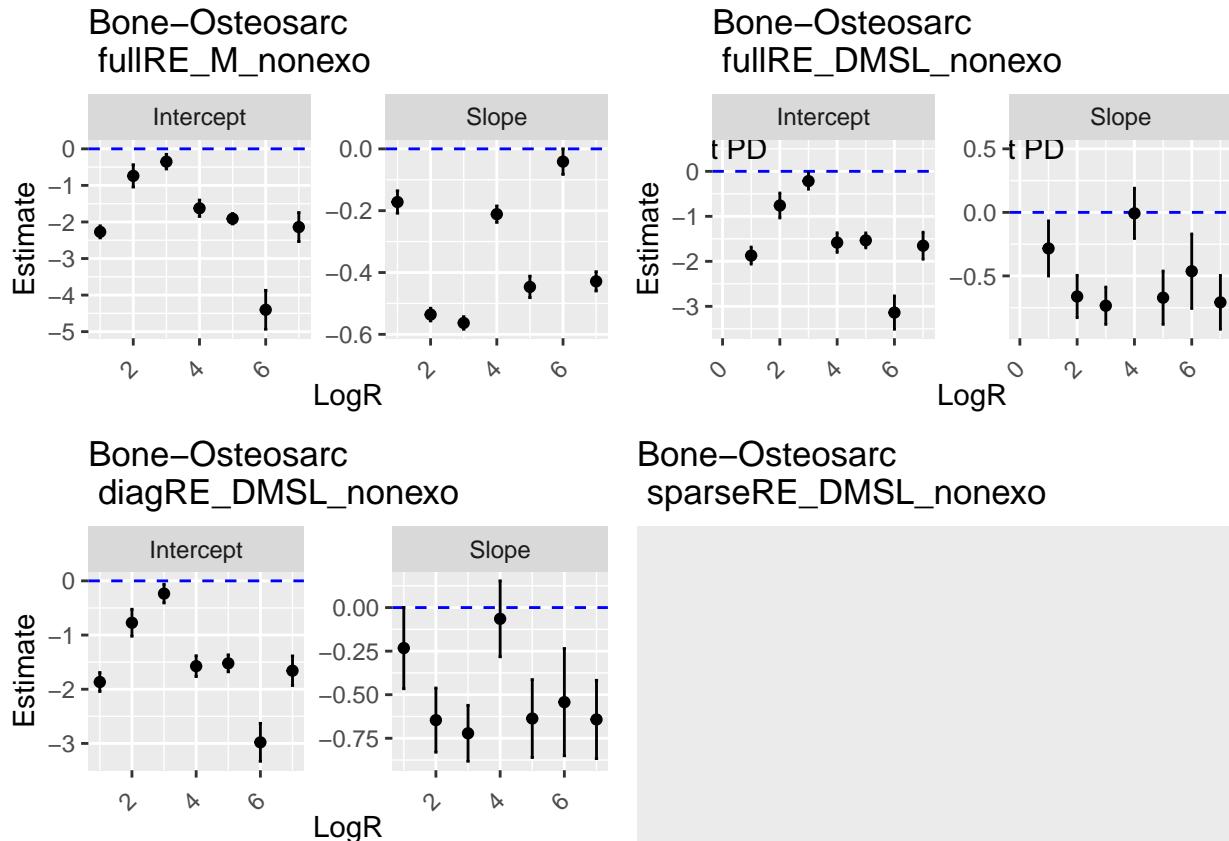


```

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 3.8923434×10^{-5} .

Covariance matrices

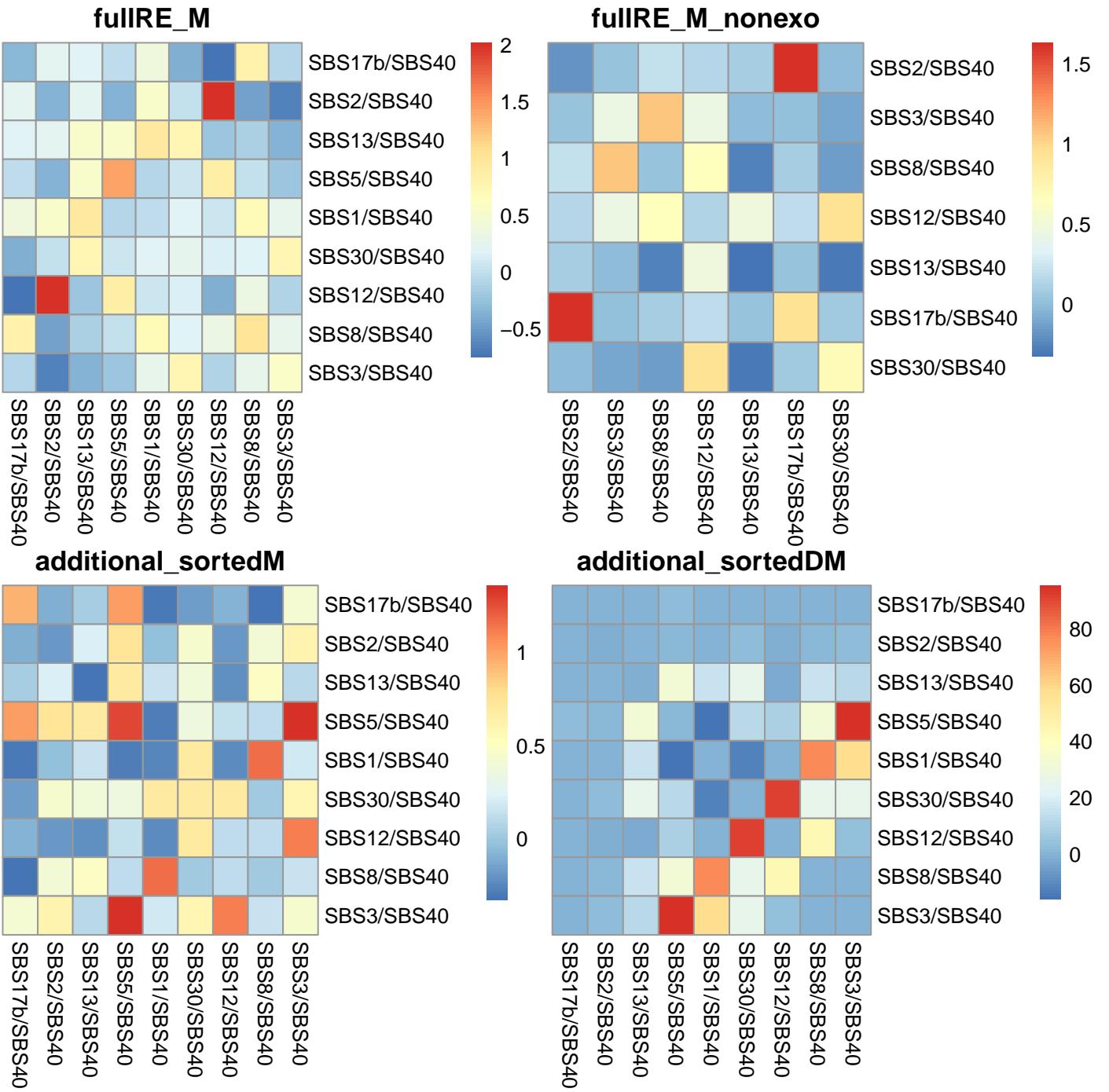
```

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

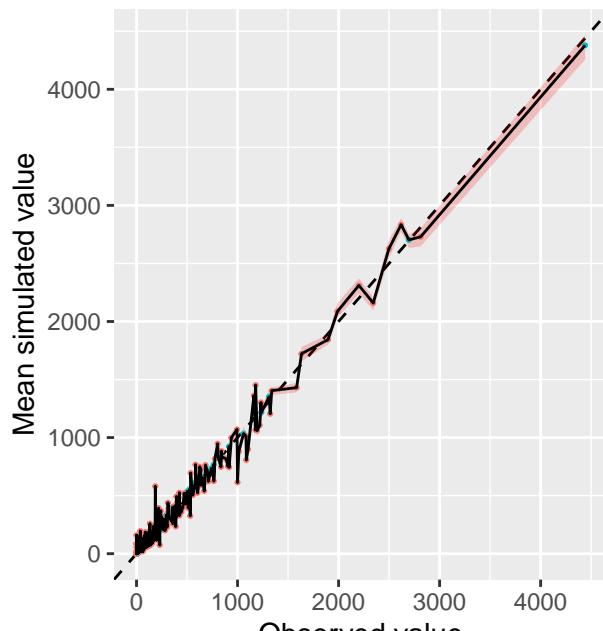
```

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)

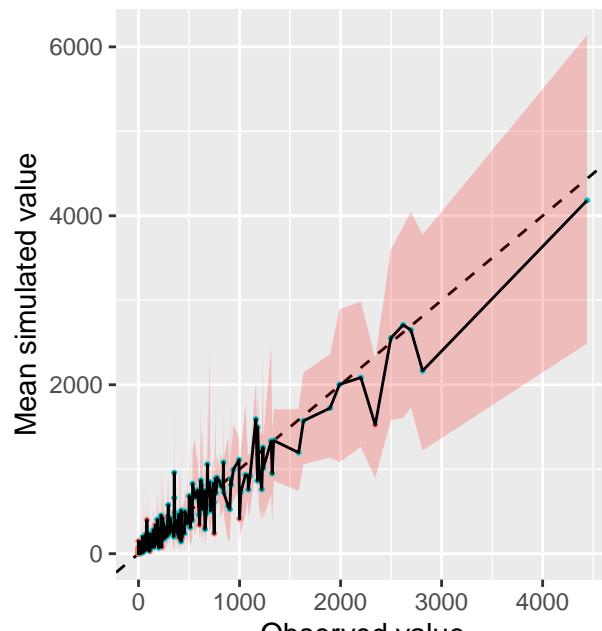
FALSE:268; TRUE:164



col • FALSE ● TRUE

obj_Bone_Osteosarc (DMSL)

FALSE:73; TRUE:359

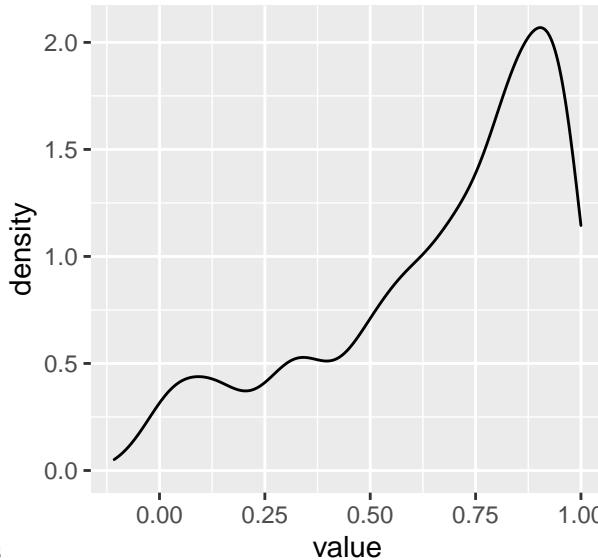


col • FALSE ● TRUE

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

Correlations of signatures

Correlations of fitted values
(fullRE M)

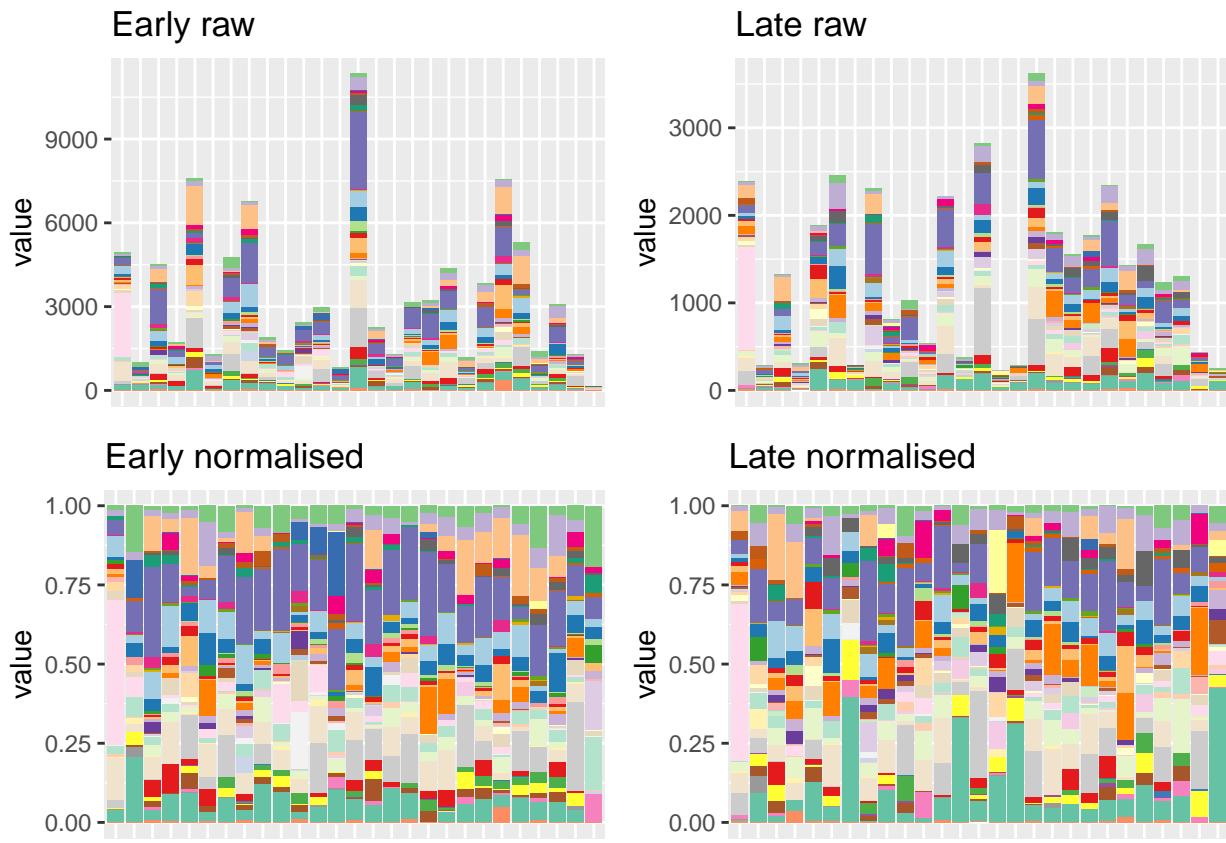


To add: the observed values, and the correlations of the normalised signatures

Signatures from mutSigExtractor

The signatures from mutSigExtractor are a bit more chaotic:

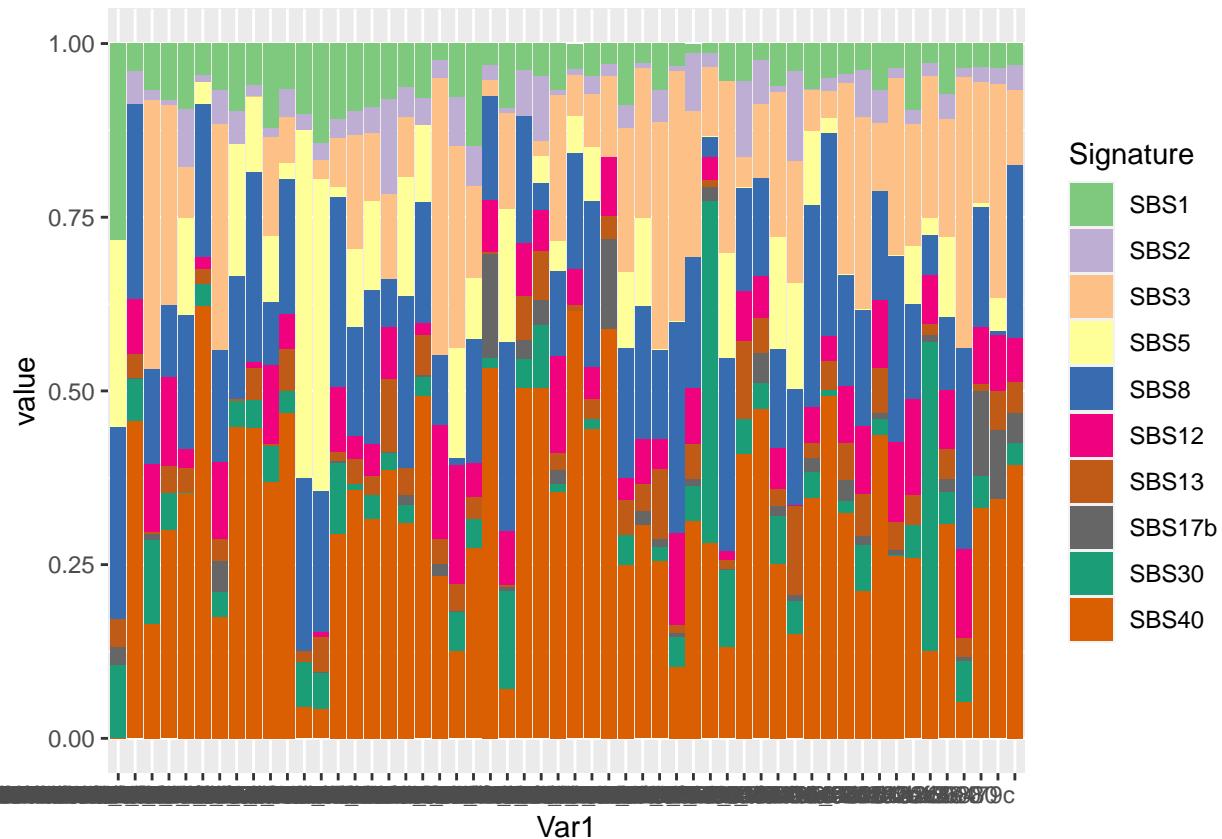
```
obj_Bone_Osteosarc_mutSigExtractor <- load_PCAWG(ct = ct, typedata = "signaturesmutSigExtractor",
                                                 path_to_data = "../..../data/")
## [1] 27
give_barplot_from_obj(obj = obj_Bone_Osteosarc_mutSigExtractor, legend_on = FALSE)
## Creating plot... it might 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)))
```

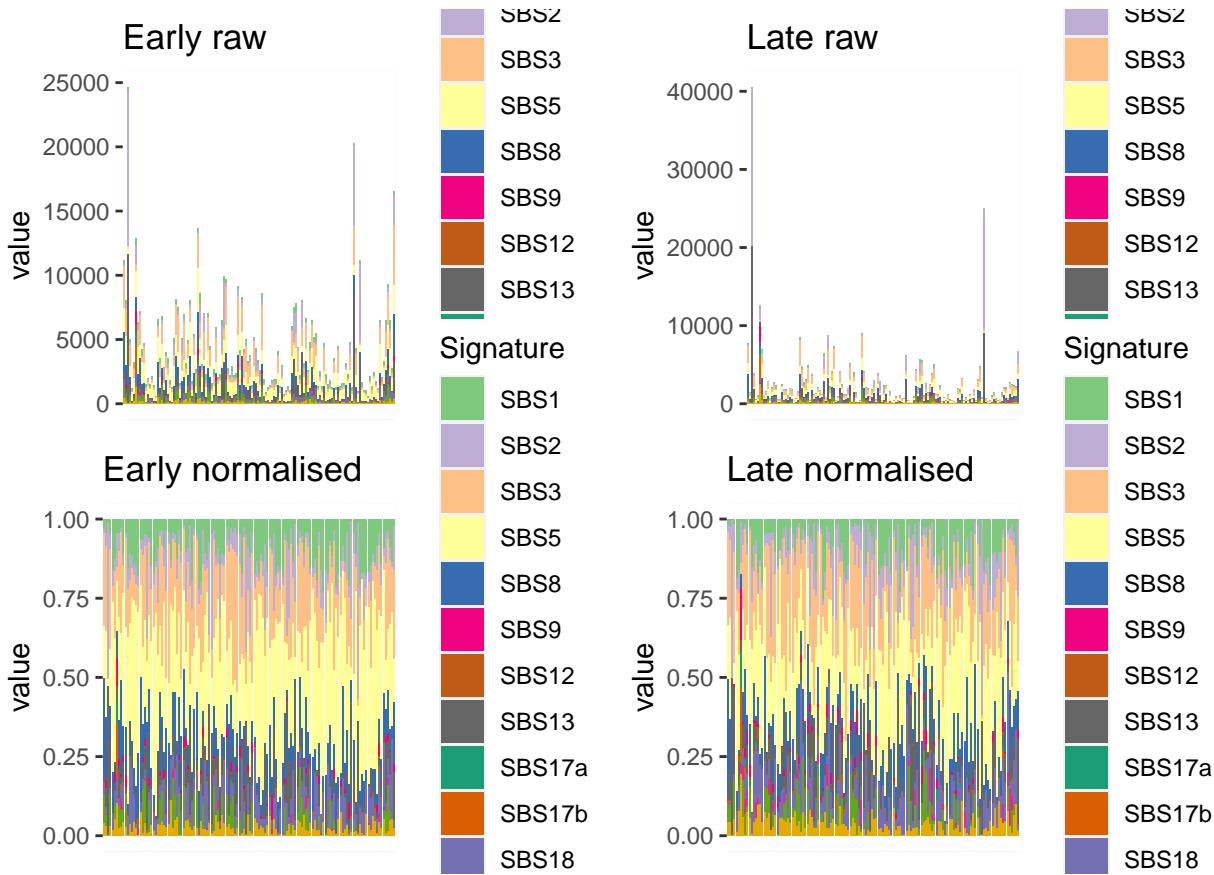
```
## Creating plot... it might take some time if the data are large. Number of samples: 54
```



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

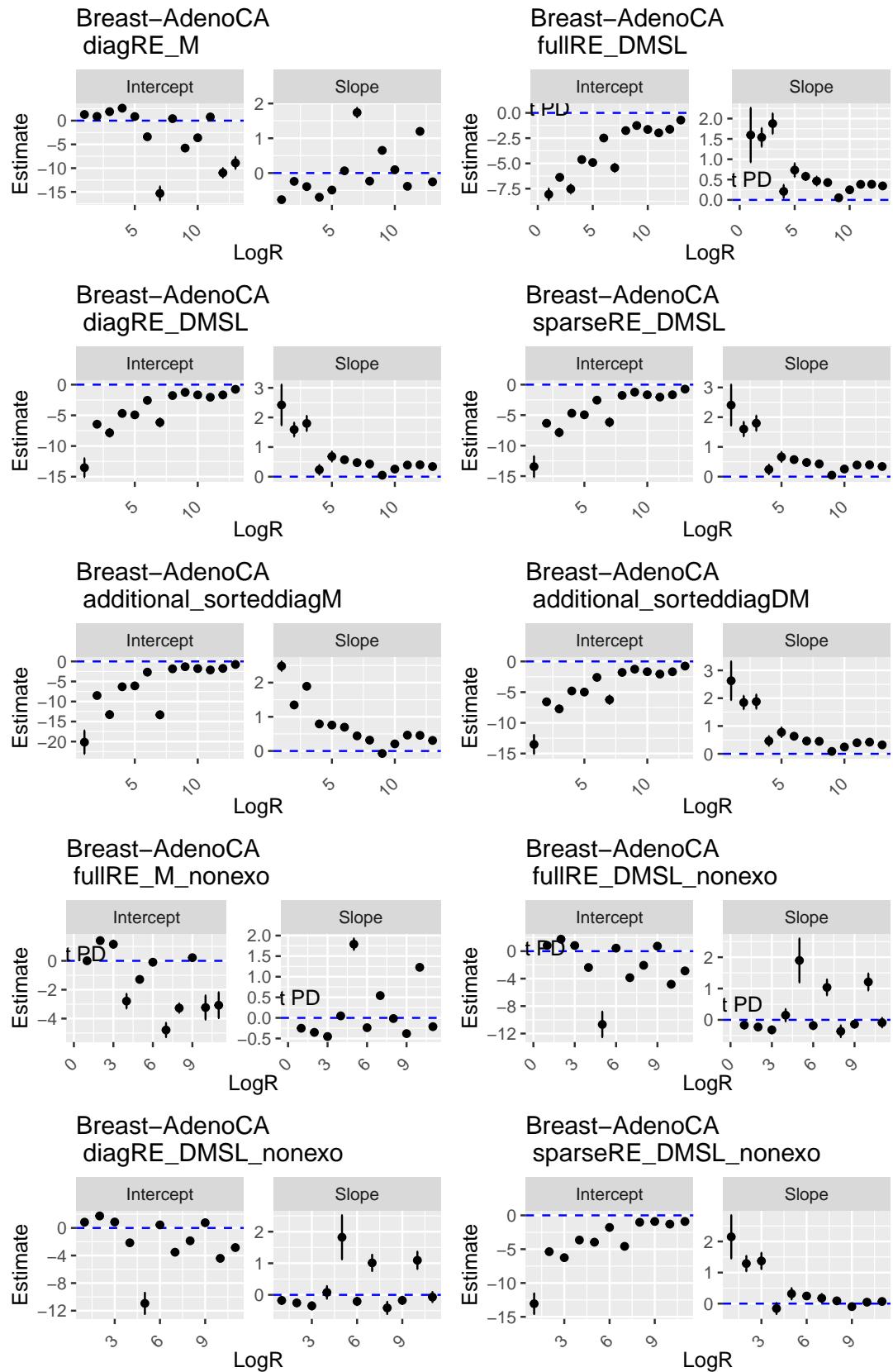
```

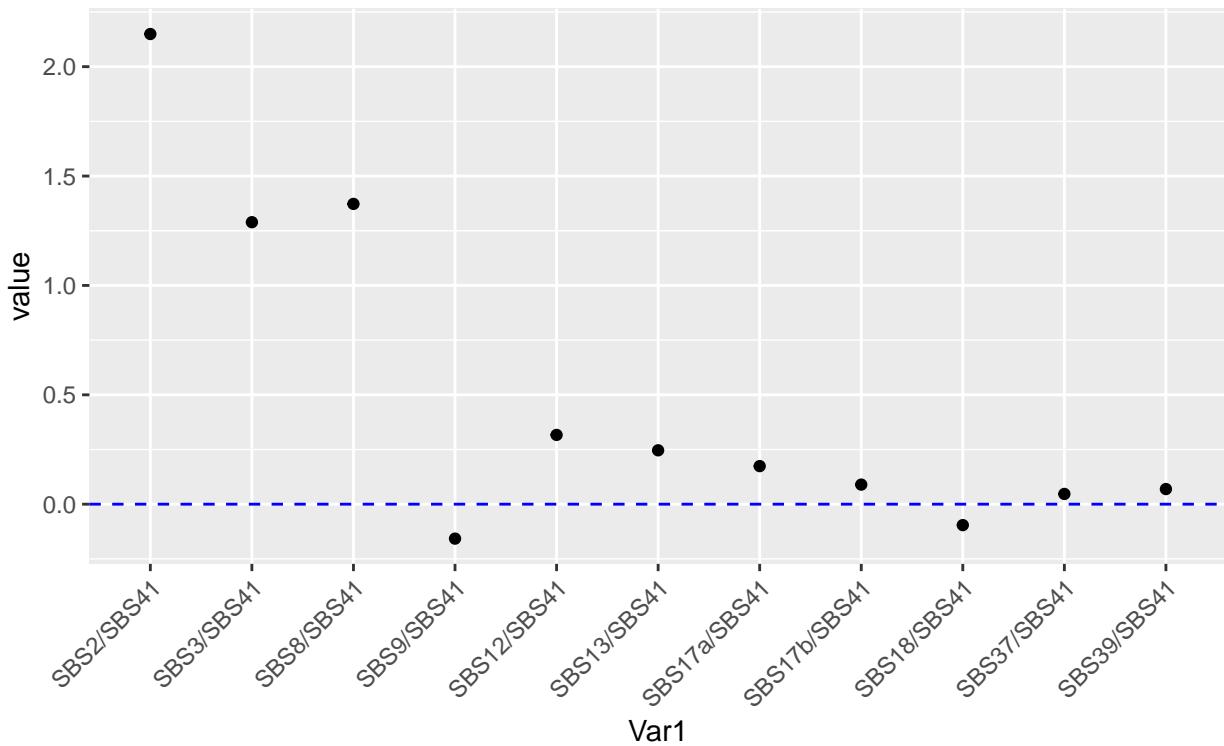
E.g.

- 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

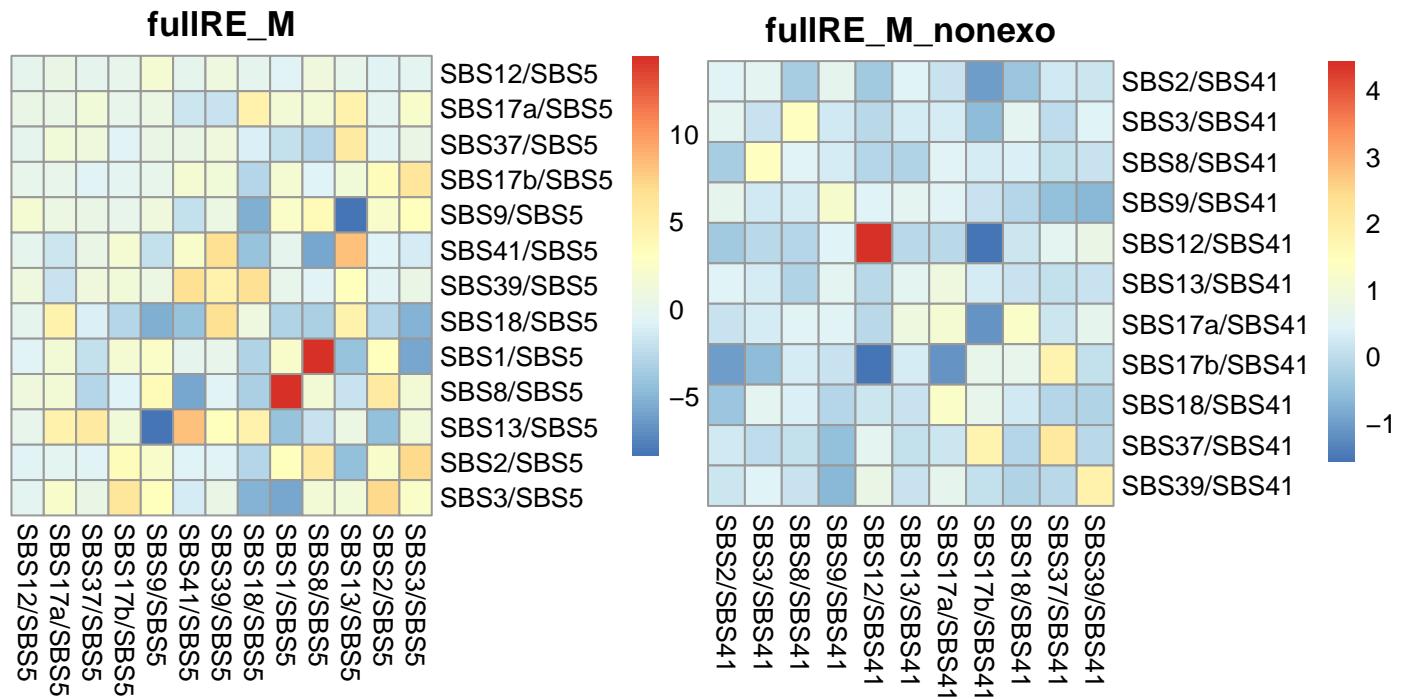
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 7.748574×10^{-12} .

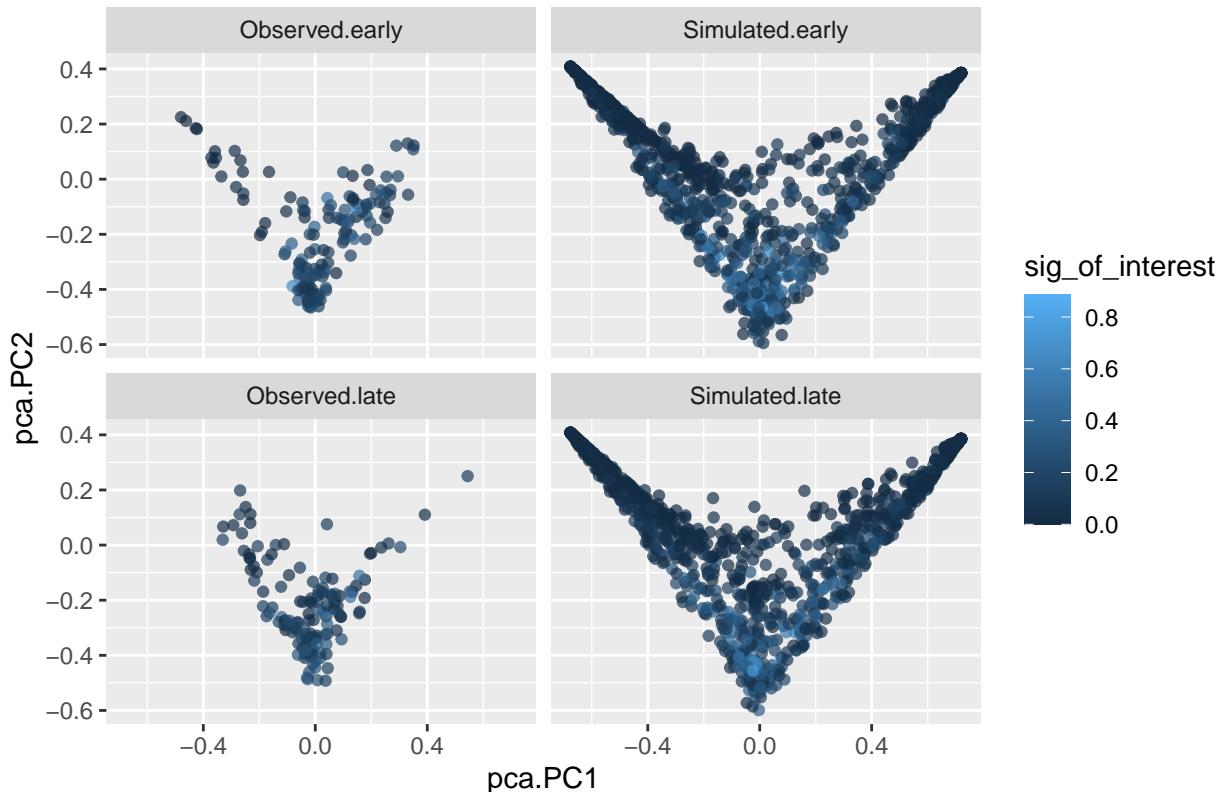
Covariance matrices



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

```

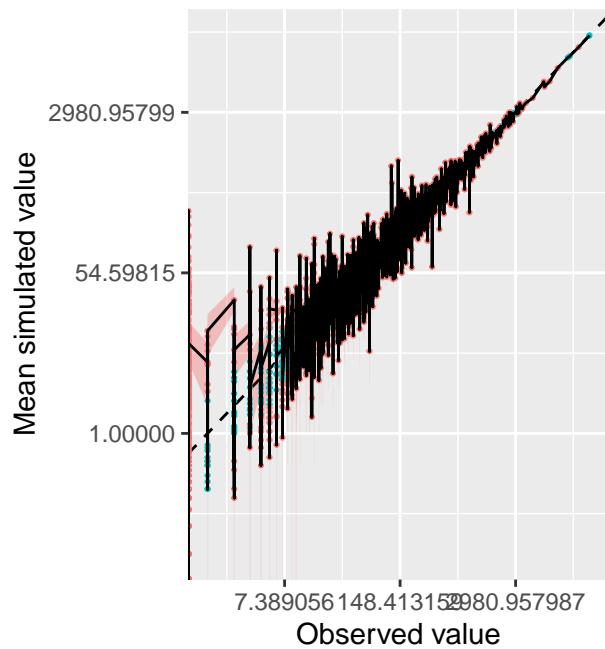
ct <- "Breast-AdenoCA"
integer_overdispersion_param_DMSL <- 1
obj_Breast_AdenoCA_nonexo <- give_subset_sigs_TMBobj(obj_Breast_AdenoCA, sigs_to_remove = nonexogenous$V1)

for(loglog_it in c(T,F)){
  grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object =
    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 = loglog_it, 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)),
    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 = loglog_it, title = 'Breast_AdenoCA_nonexo (DMSL)', ncol=2)
}
## Warning: Transformation introduced infinite values in continuous y-axis

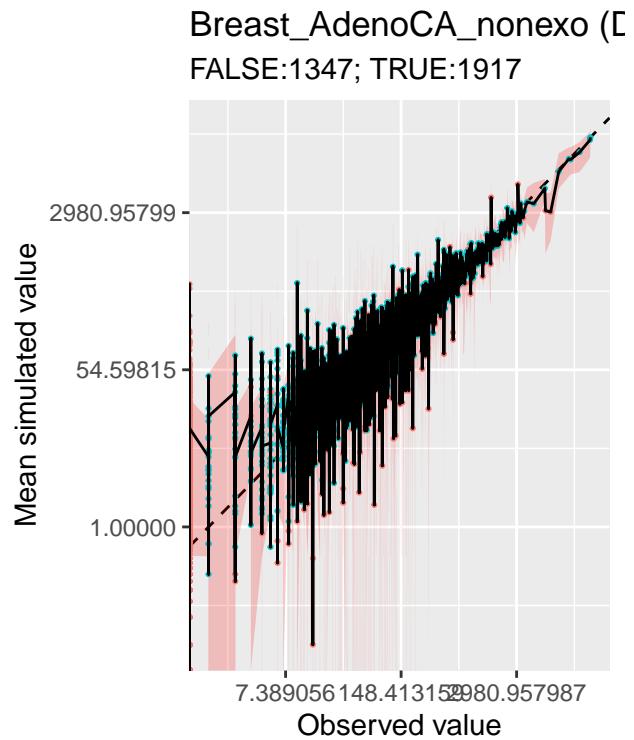
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
```

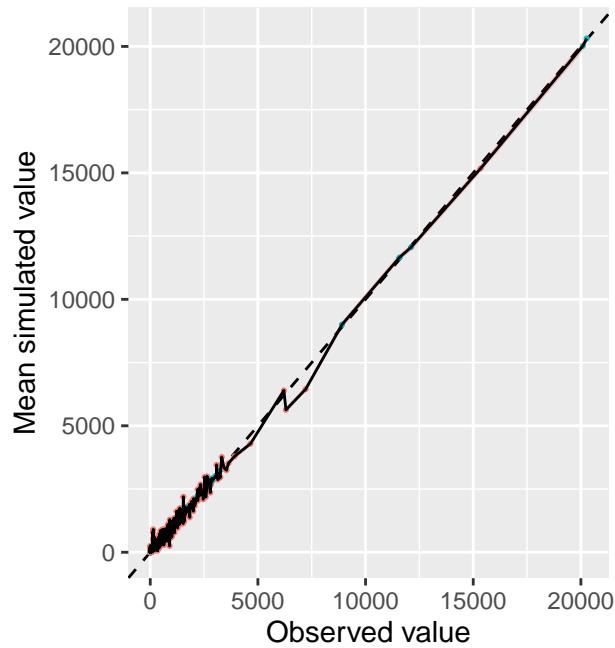
Breast_AdenoCA_nonexo (A)
FALSE:2396; TRUE:868



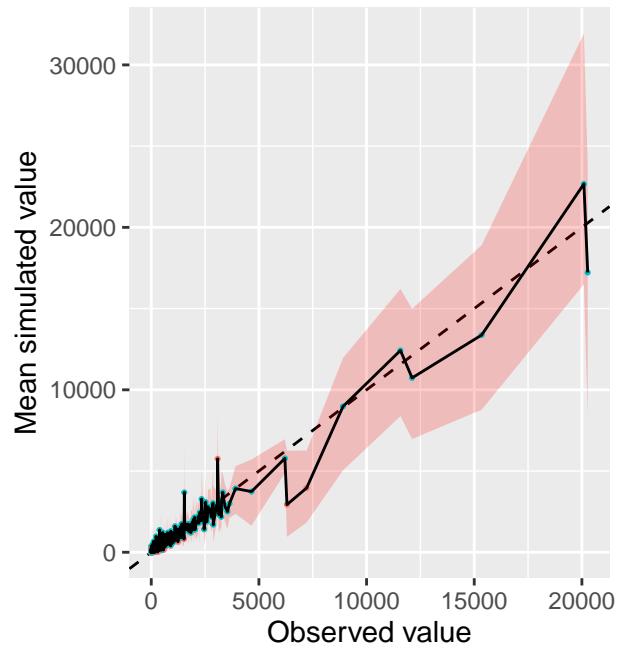
Breast_AdenoCA_nonexo (E)
FALSE:1347; TRUE:1917



Breast_AdenoCA_nonexo (M)
FALSE:2417; TRUE:847



Breast_AdenoCA_nonexo (DMS)
FALSE:1337; TRUE:1927



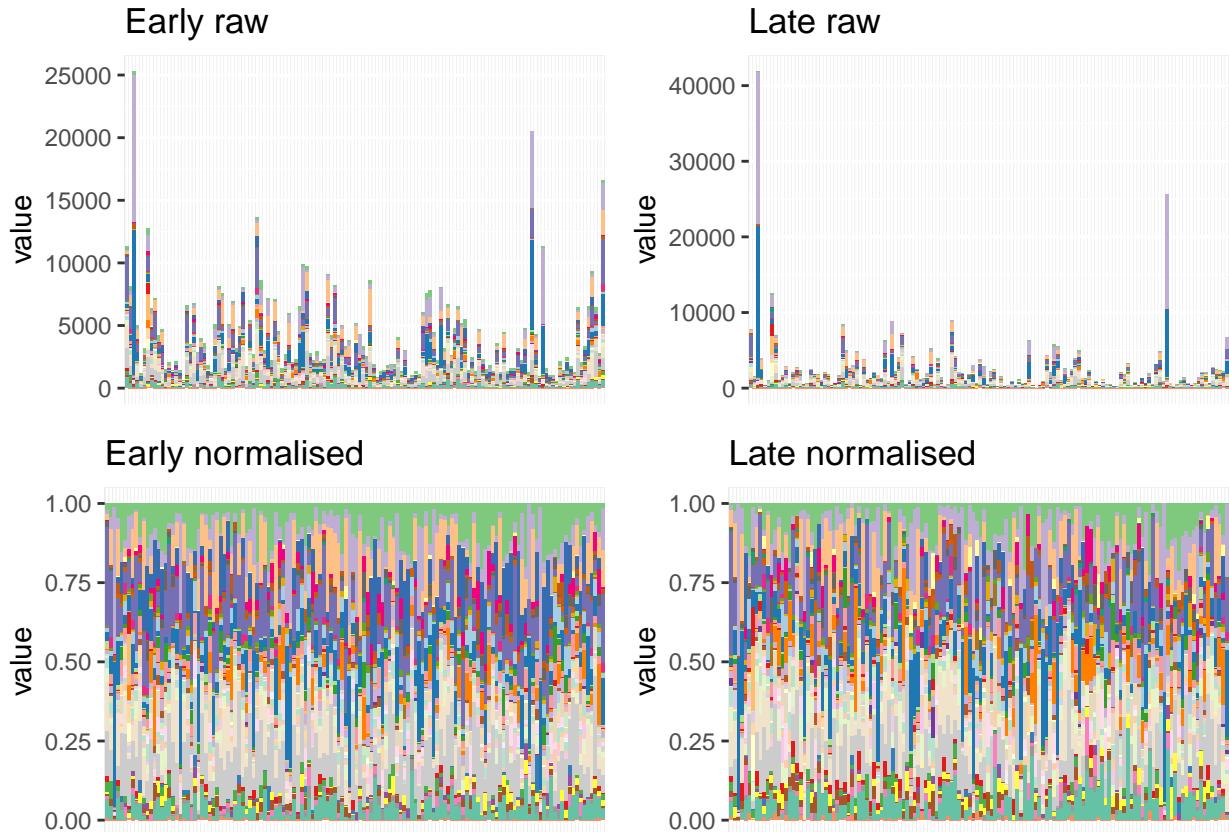
Signatures from mutSigExtractor

```
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
```

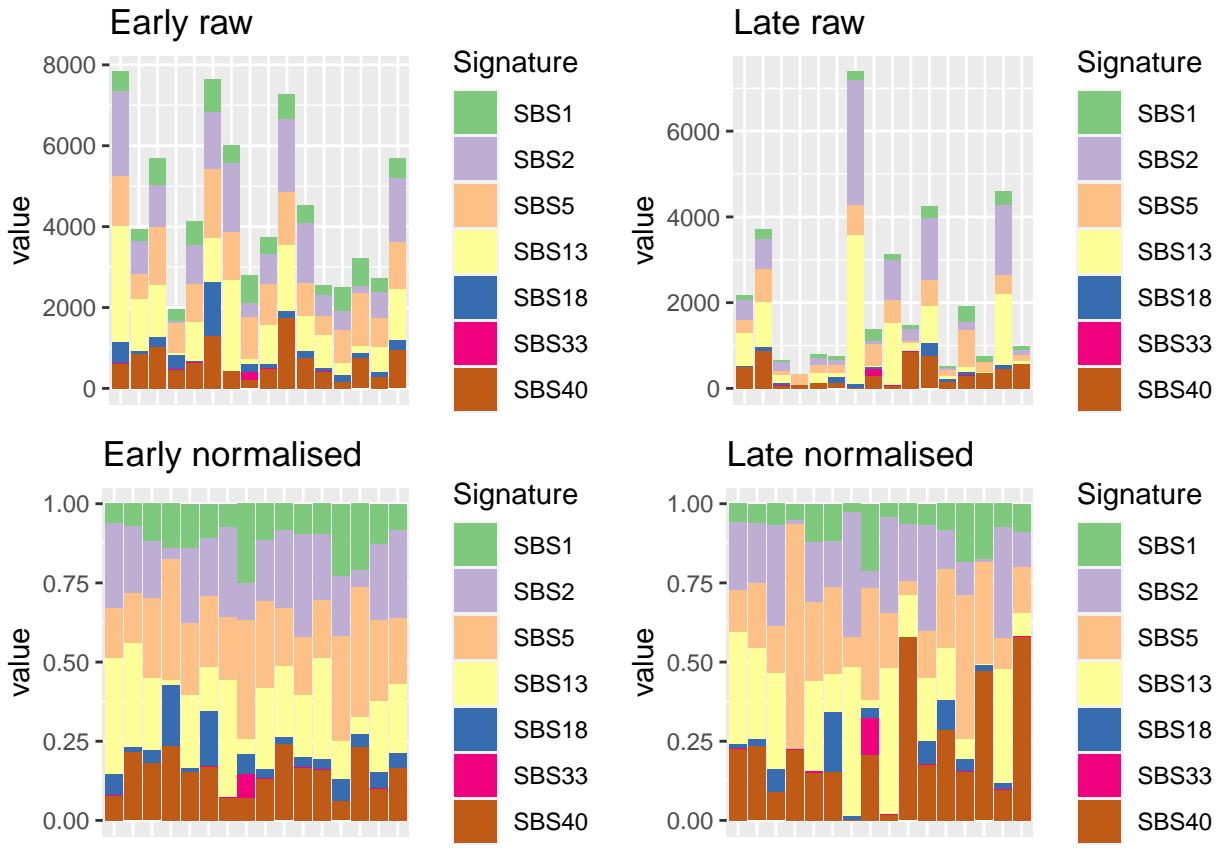


Cervix-SCC

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)

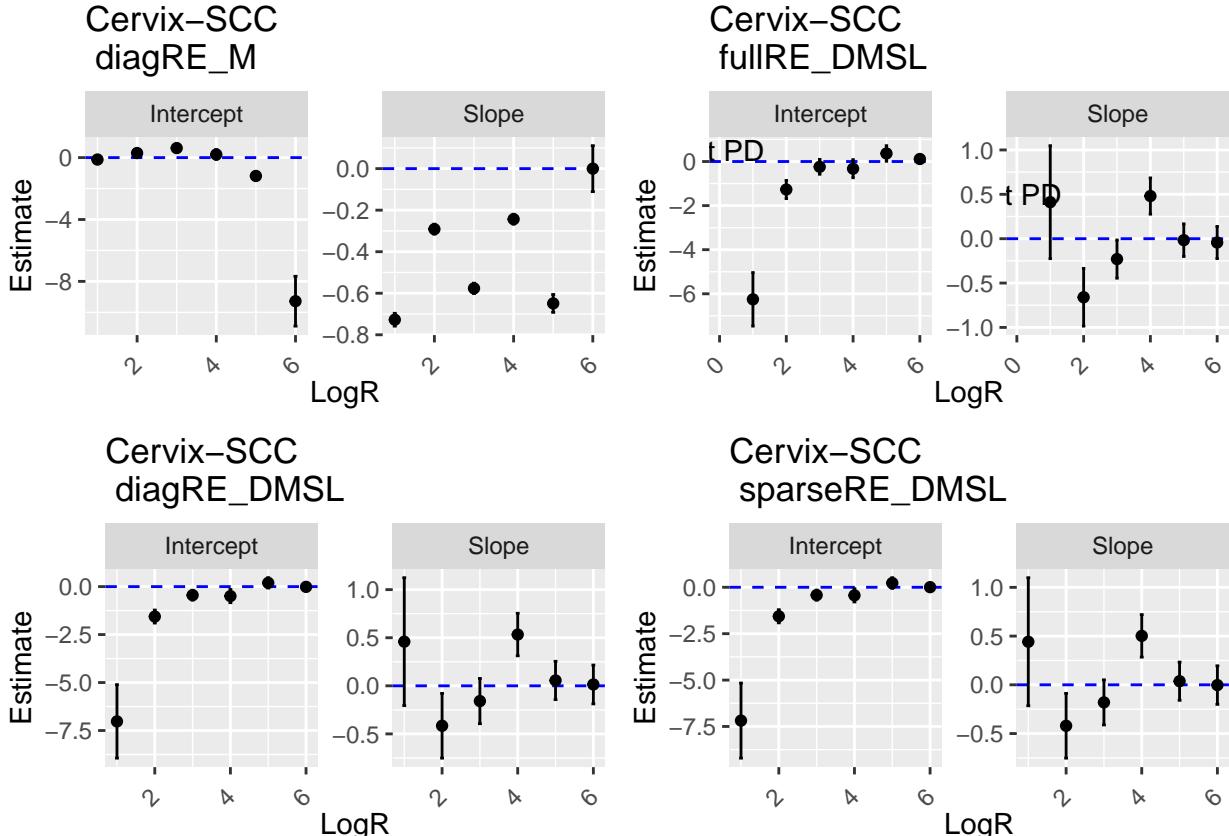
##      SBS1      SBS2      SBS5     SBS13     SBS18     SBS33
## 0.099164517 0.235000561 0.211562185 0.250439236 0.046577698 0.003560615
##      SBS40
## 0.153695189
```

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)
```

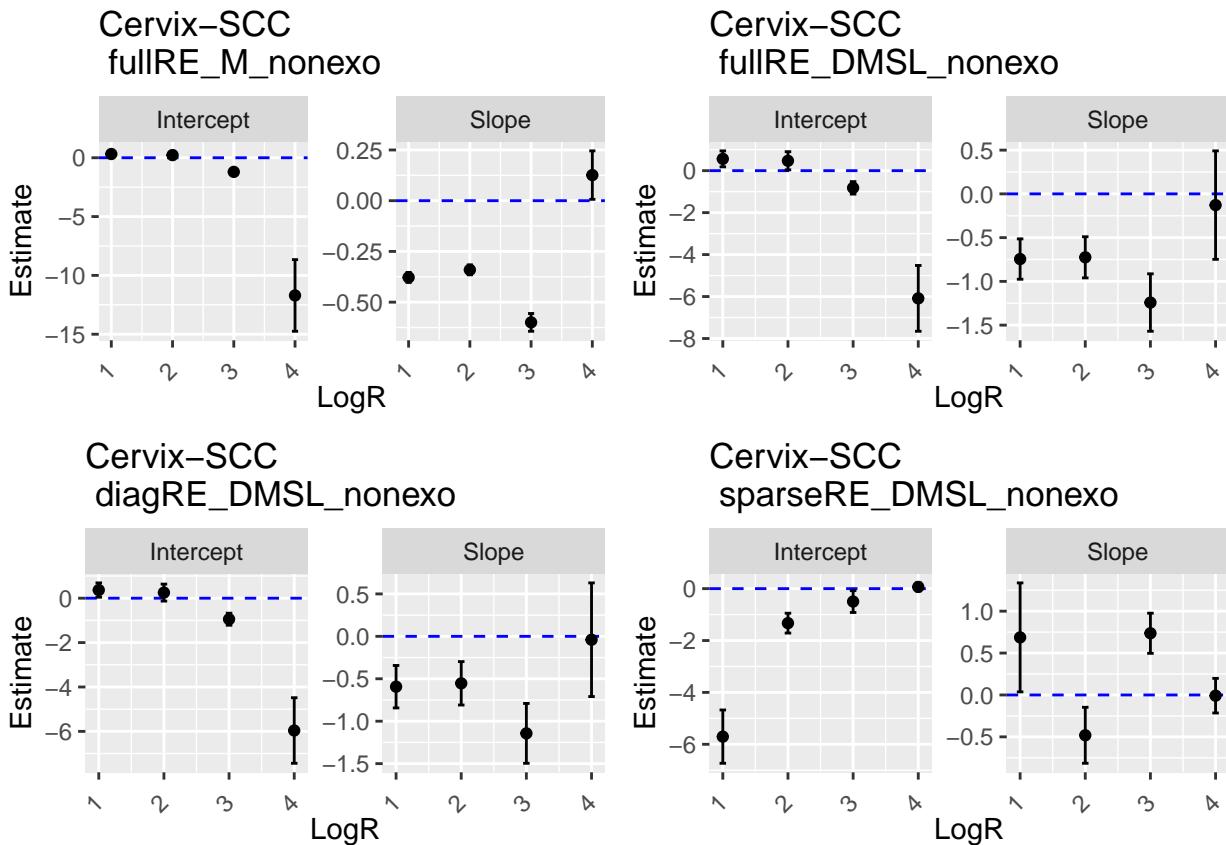
```
## Warning in sqrt(diag(object$cov.fixed)): NaNs produced
```



```

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 3.8923434×10^{-5} .

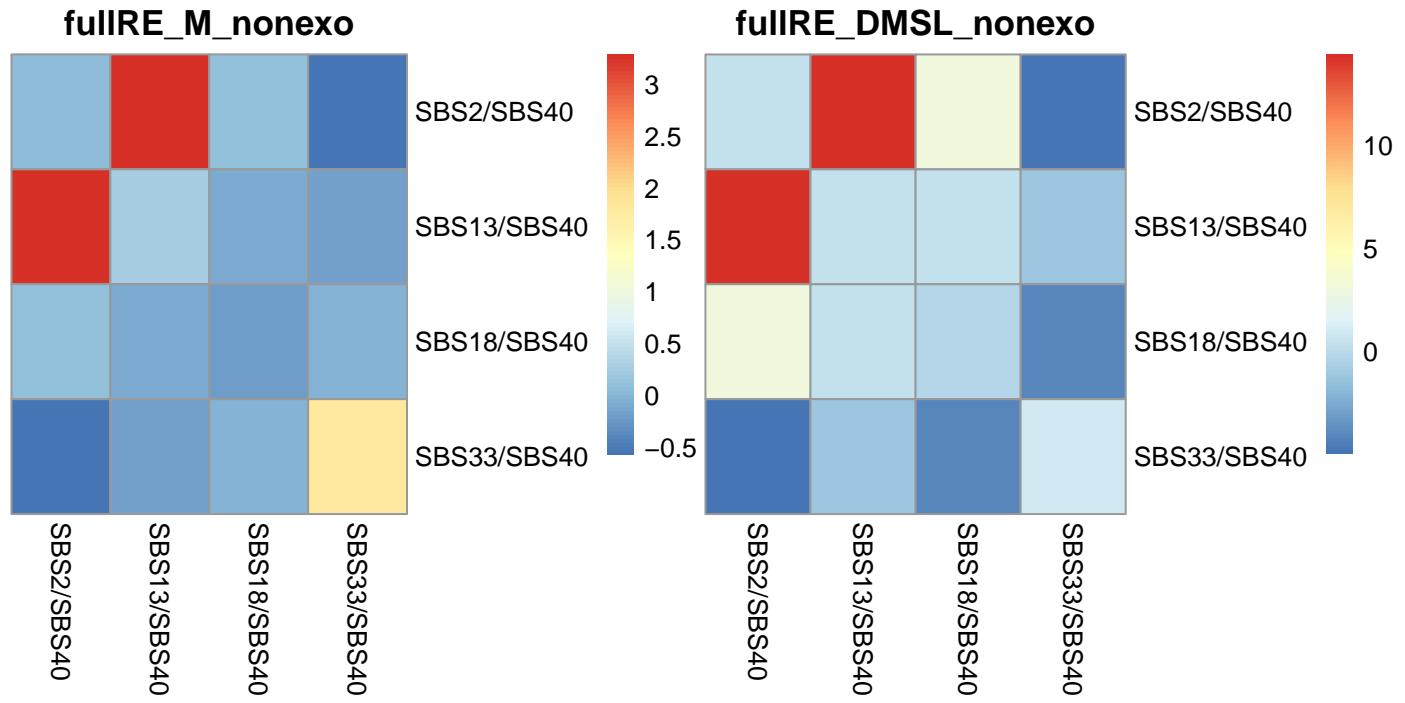
Covariance matrices

```

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

```

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



Simulation under inferred data

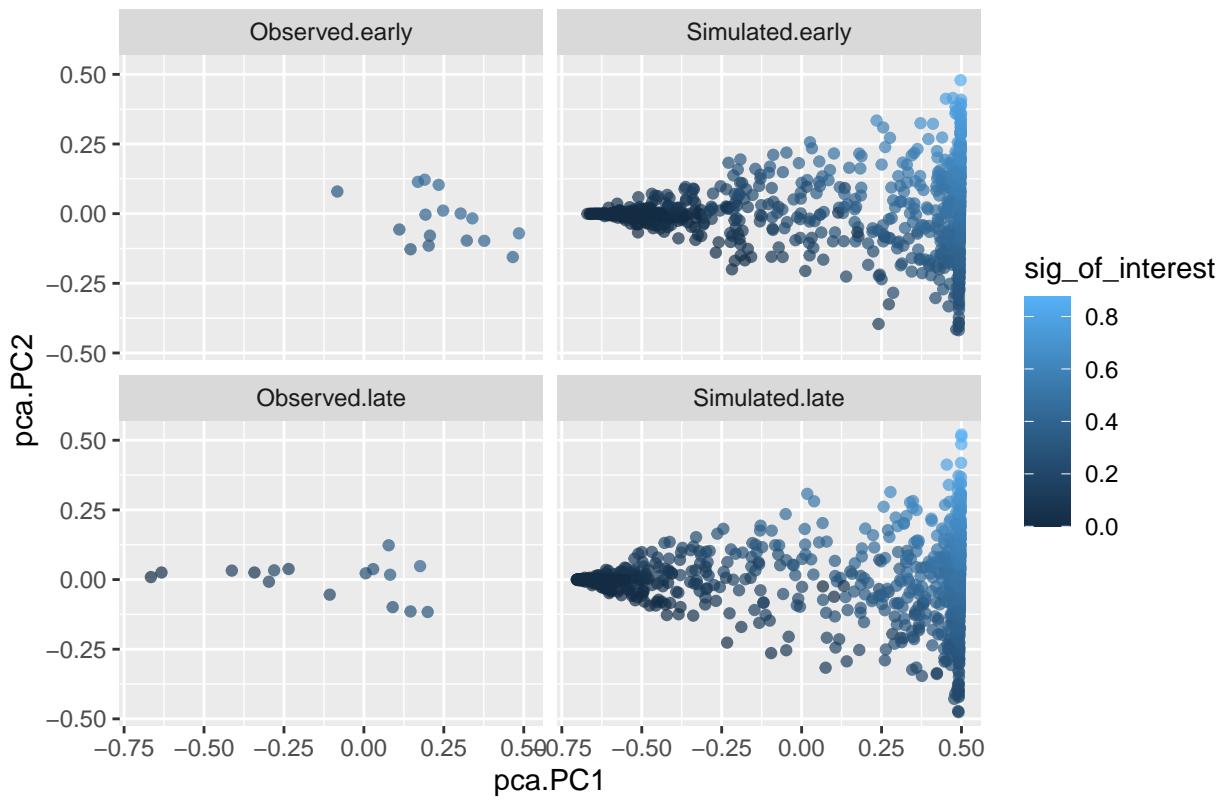
```
unique(nonexogenous$V1)

## [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

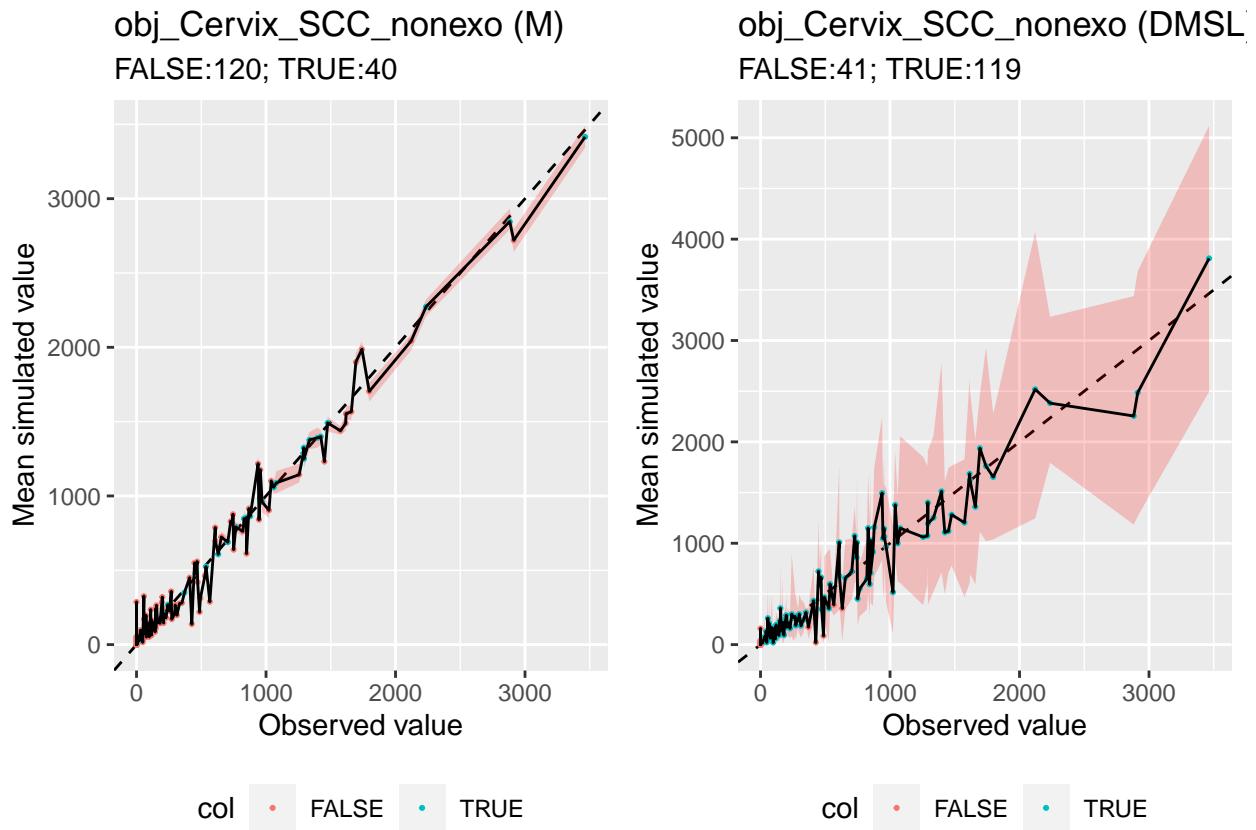
## Warning in mvtnorm:::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

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)
```



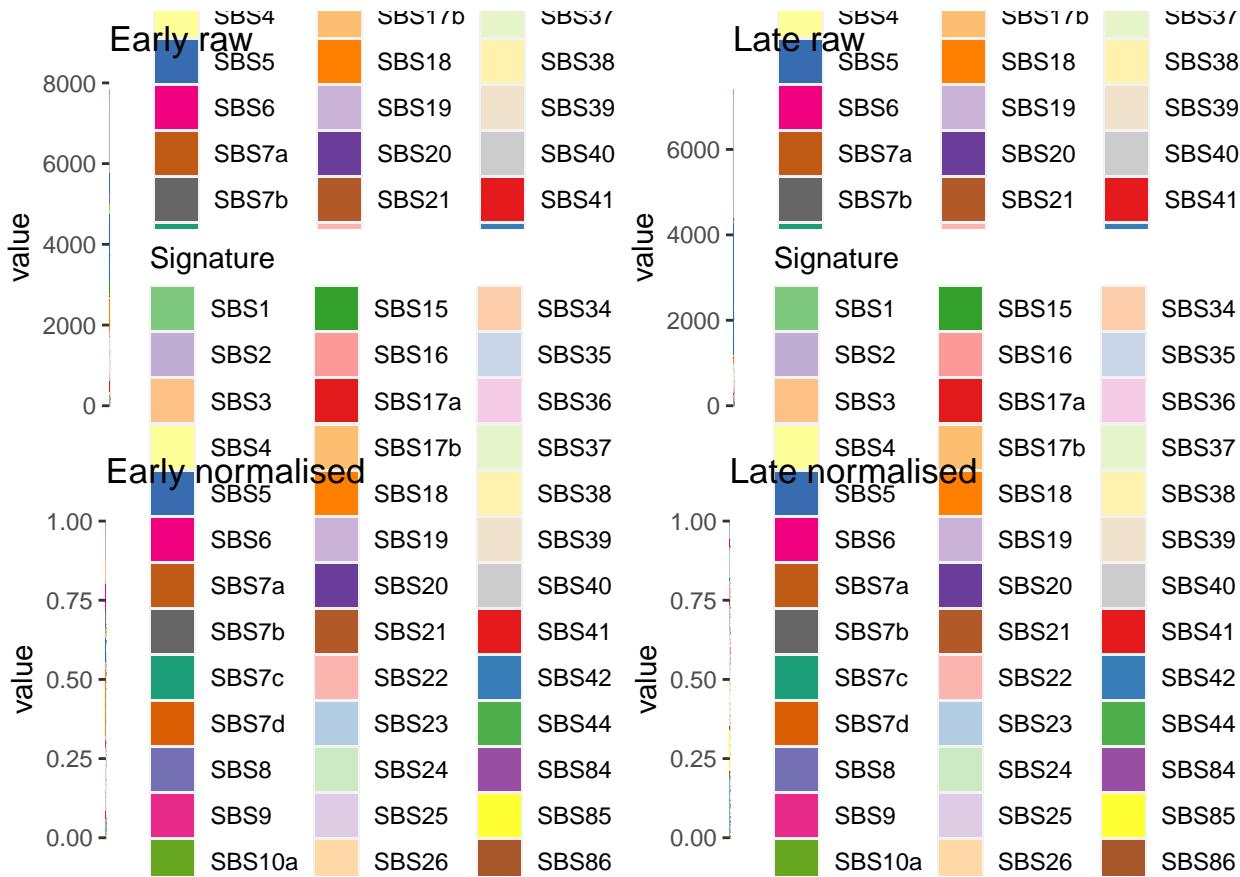
Signatures from mutSigExtractor

```
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)

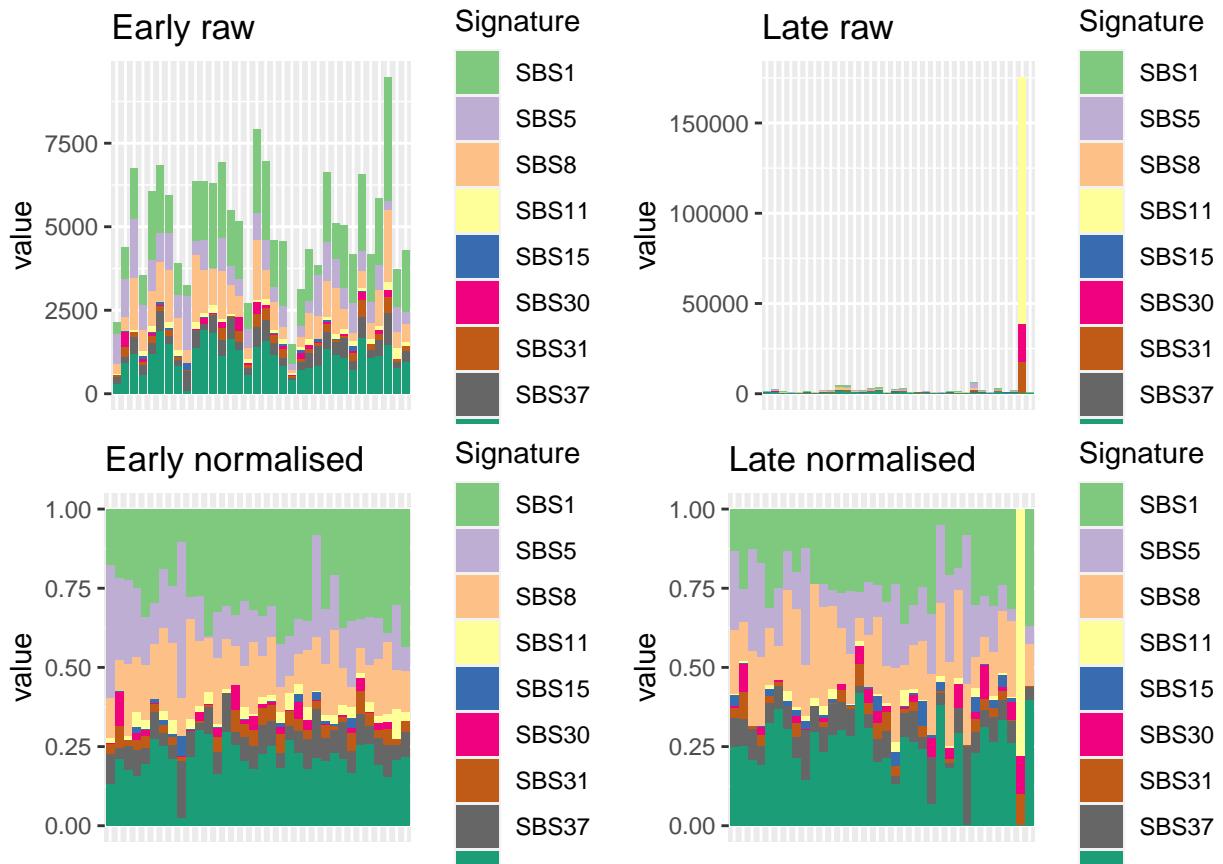
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take some time if the data are large. Number of samples: 16
## Creating plot... it might take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

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

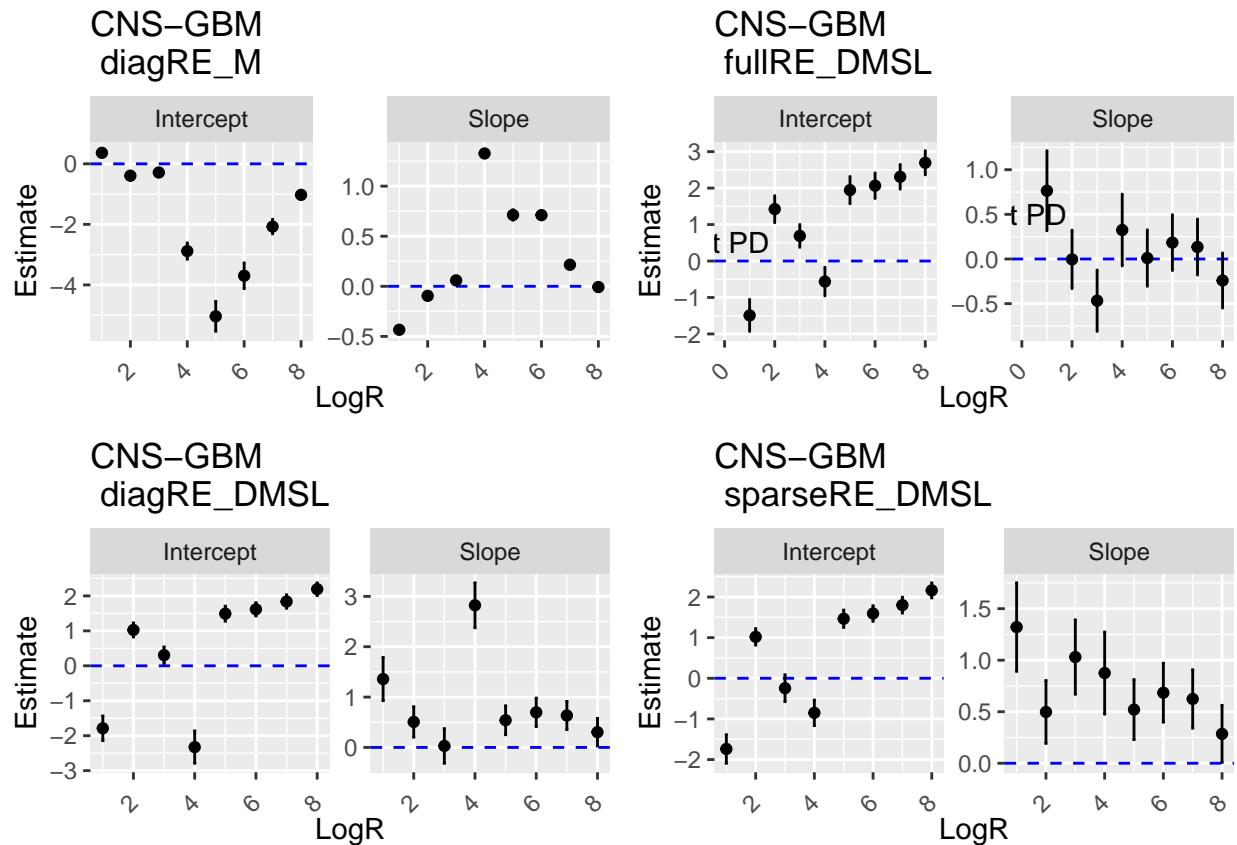
```

ct <- "CNS-GBM"

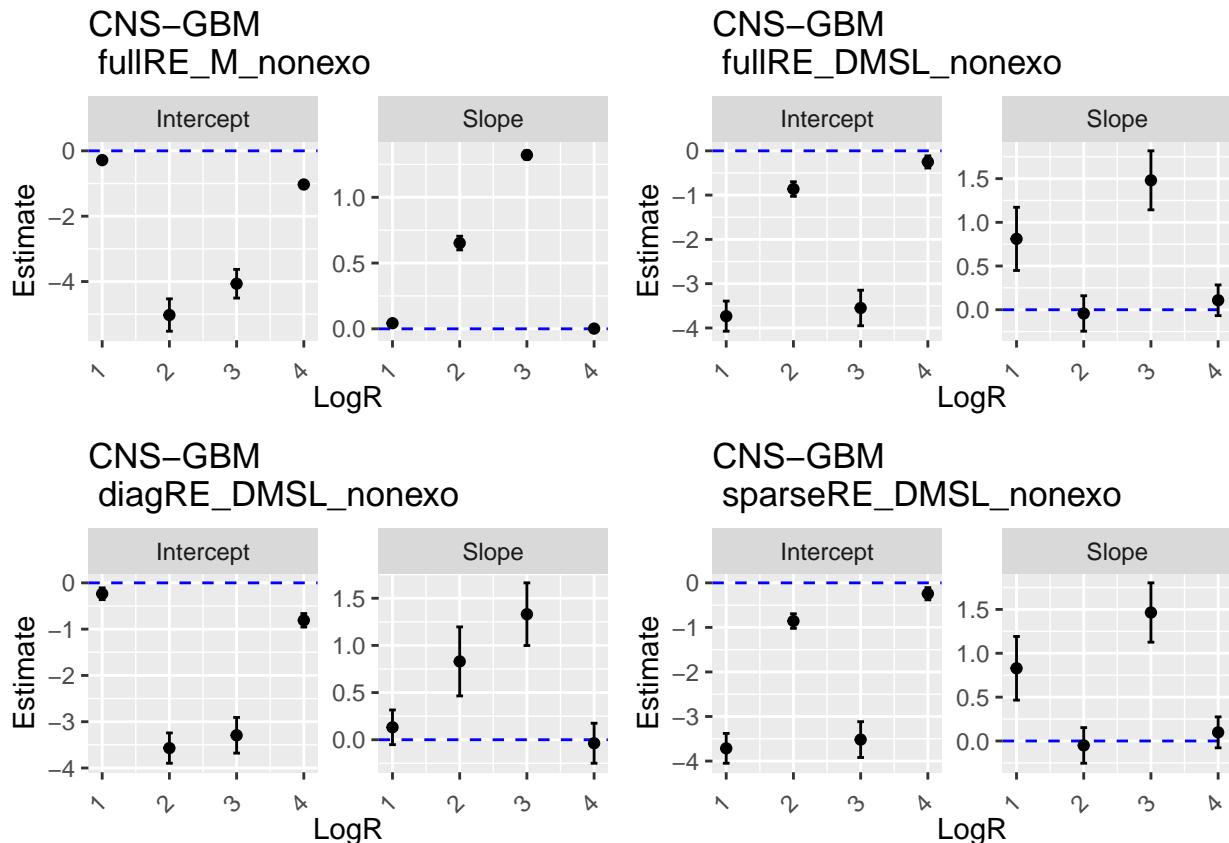
grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)

## 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

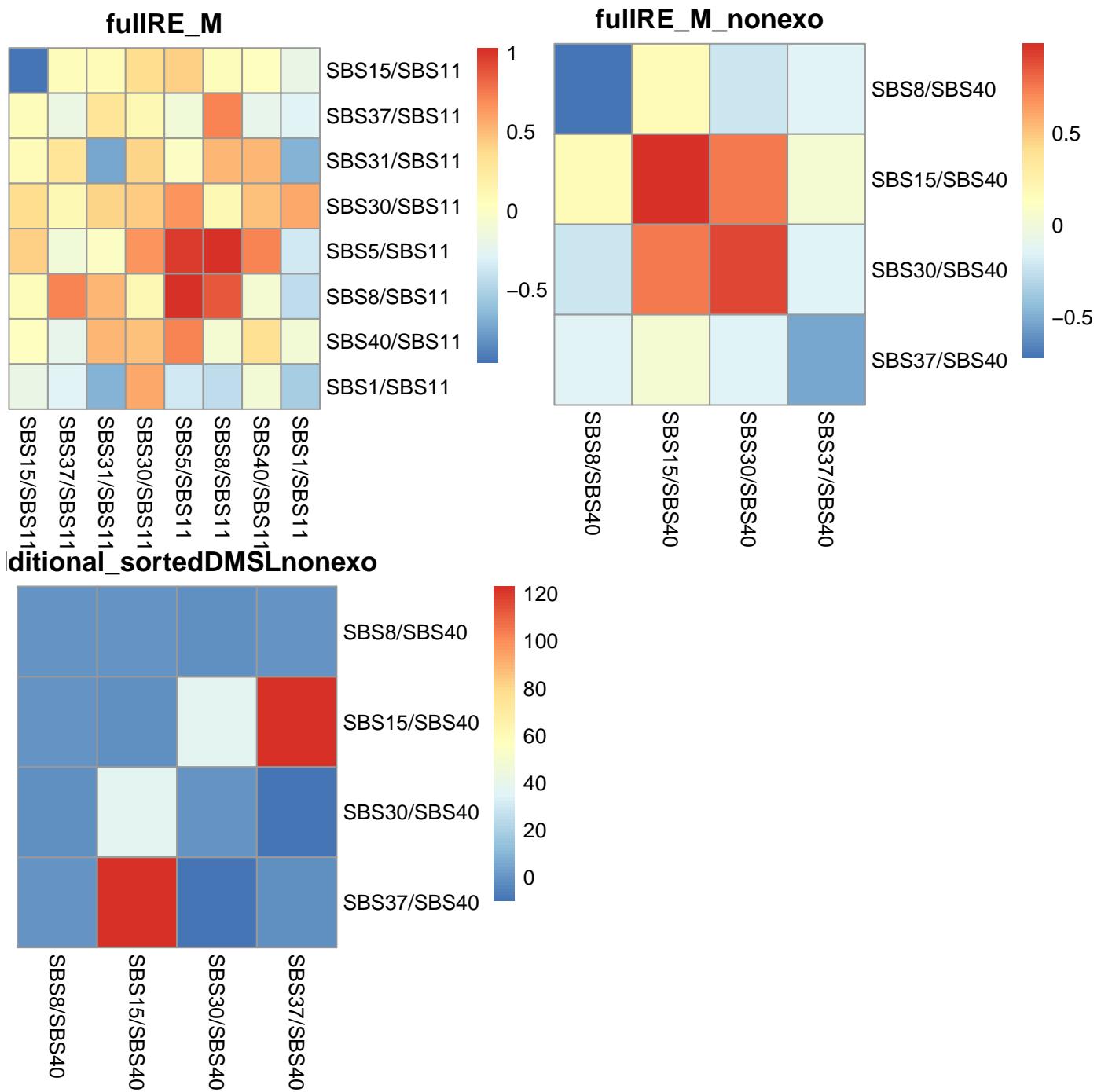
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 6.6827492×10^{-5} .

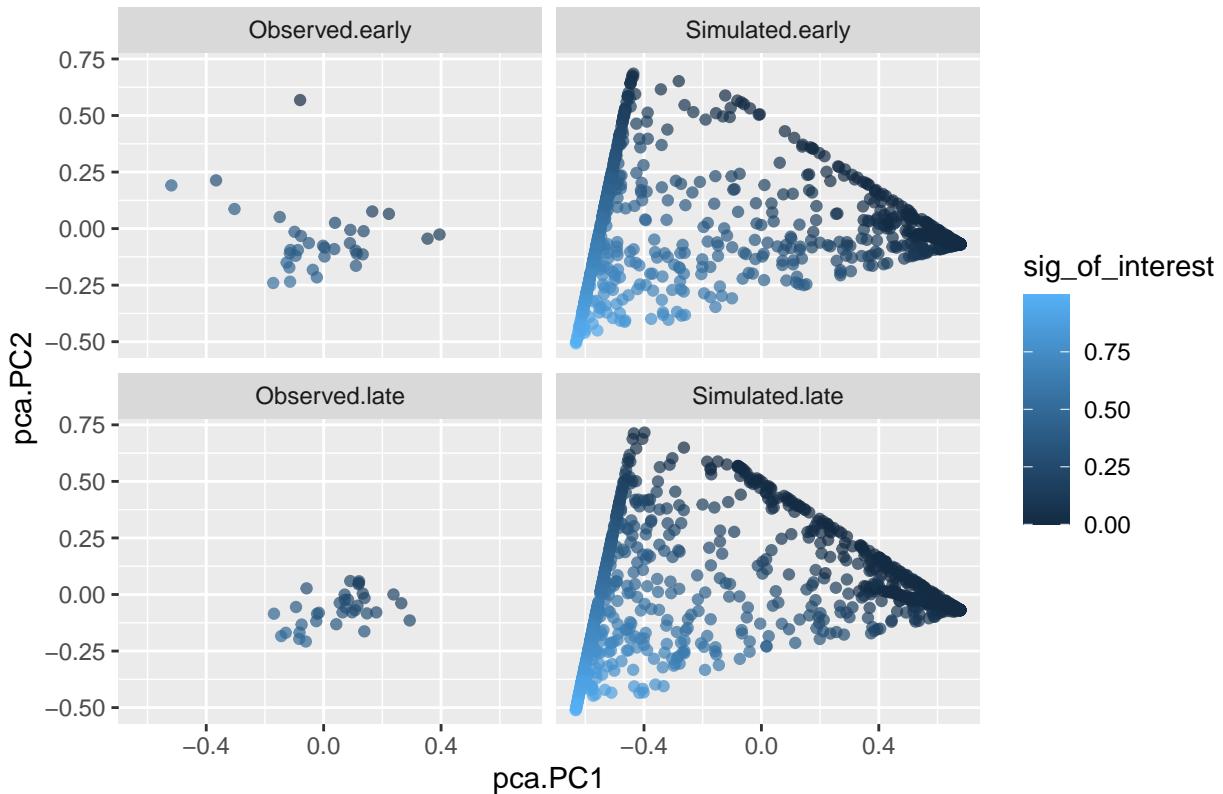
Covariance matrices



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

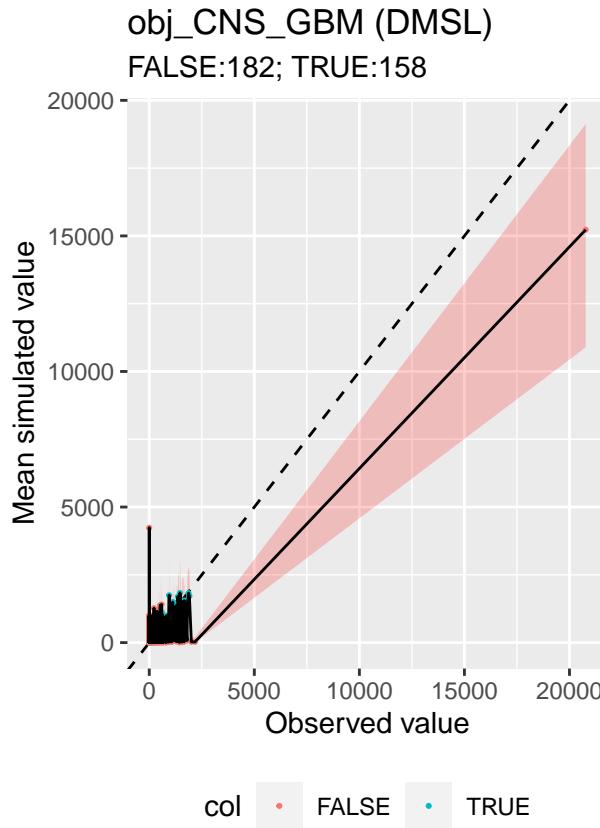
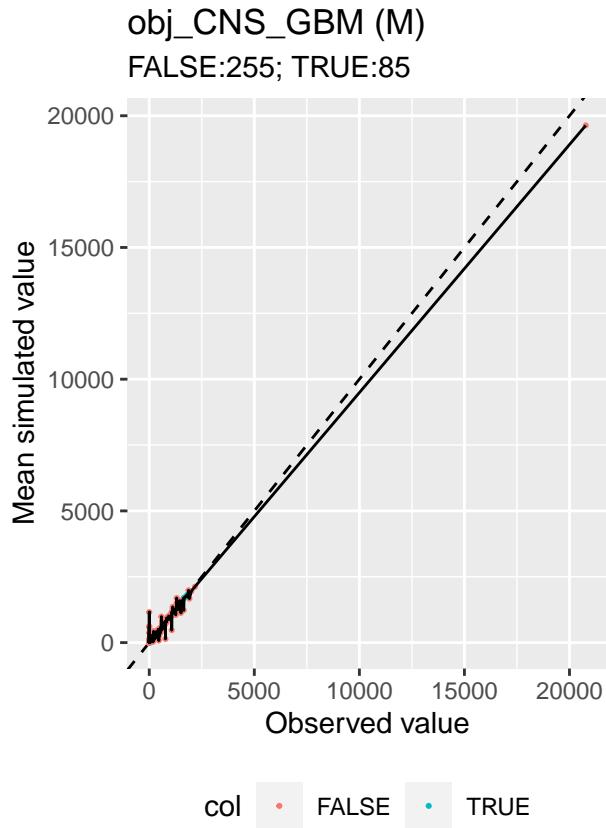


Ranked plot for coverage

```

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

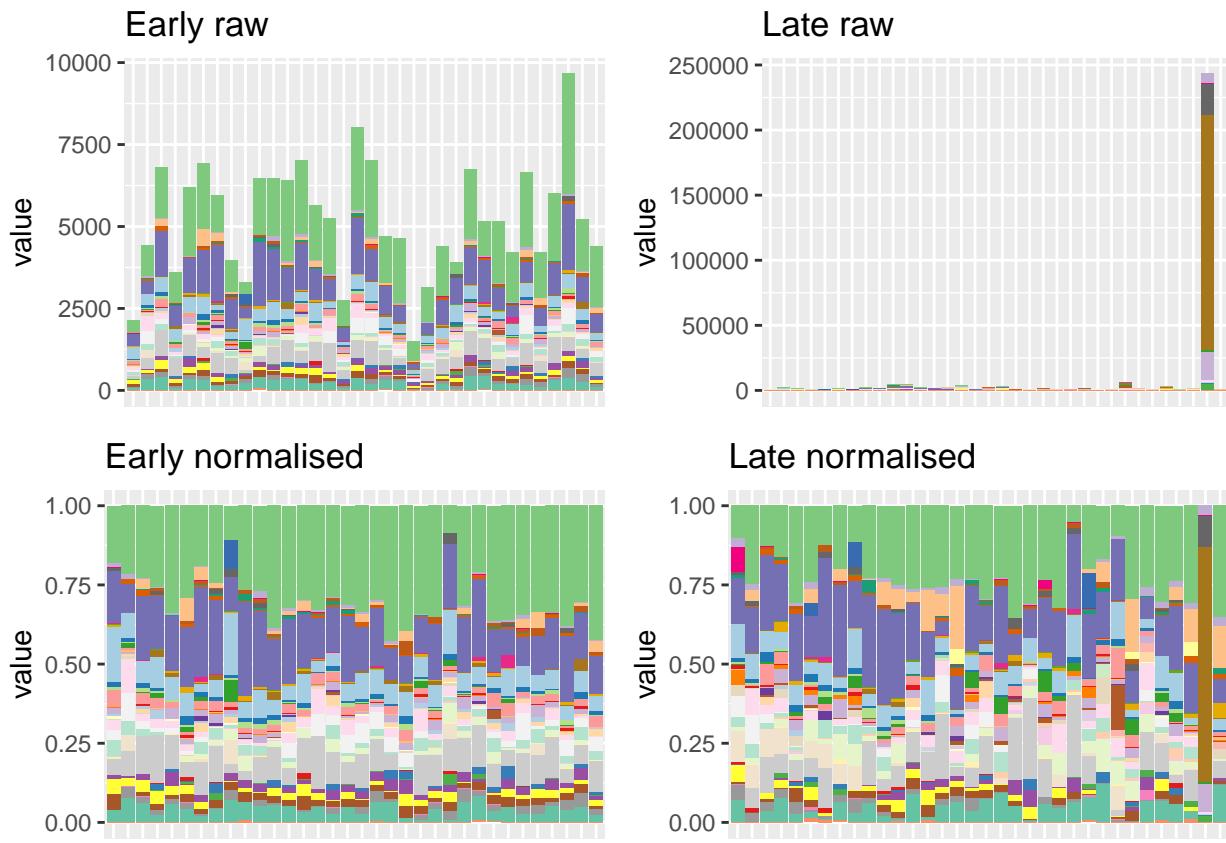
Signatures from mutSigExtractor

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)

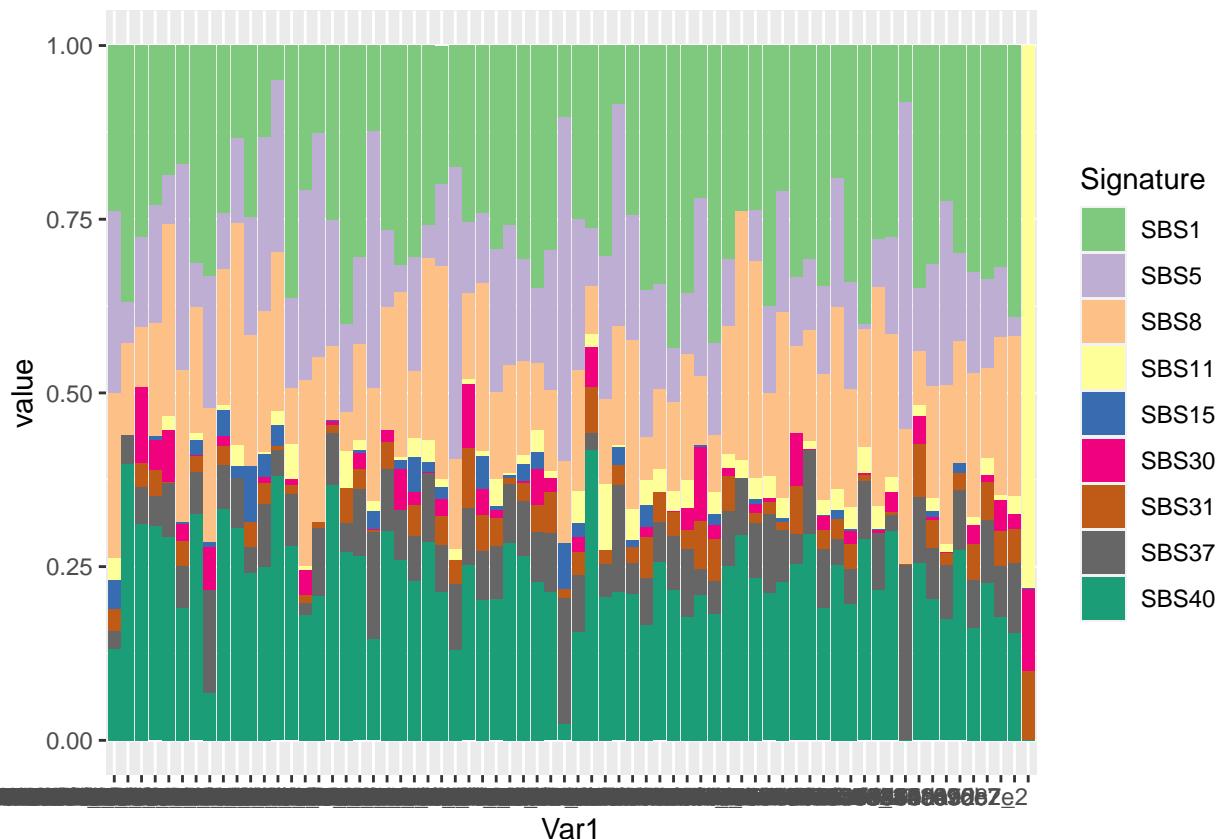
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take some time if the data are large. Number of samples: 34
## Creating plot... it might take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

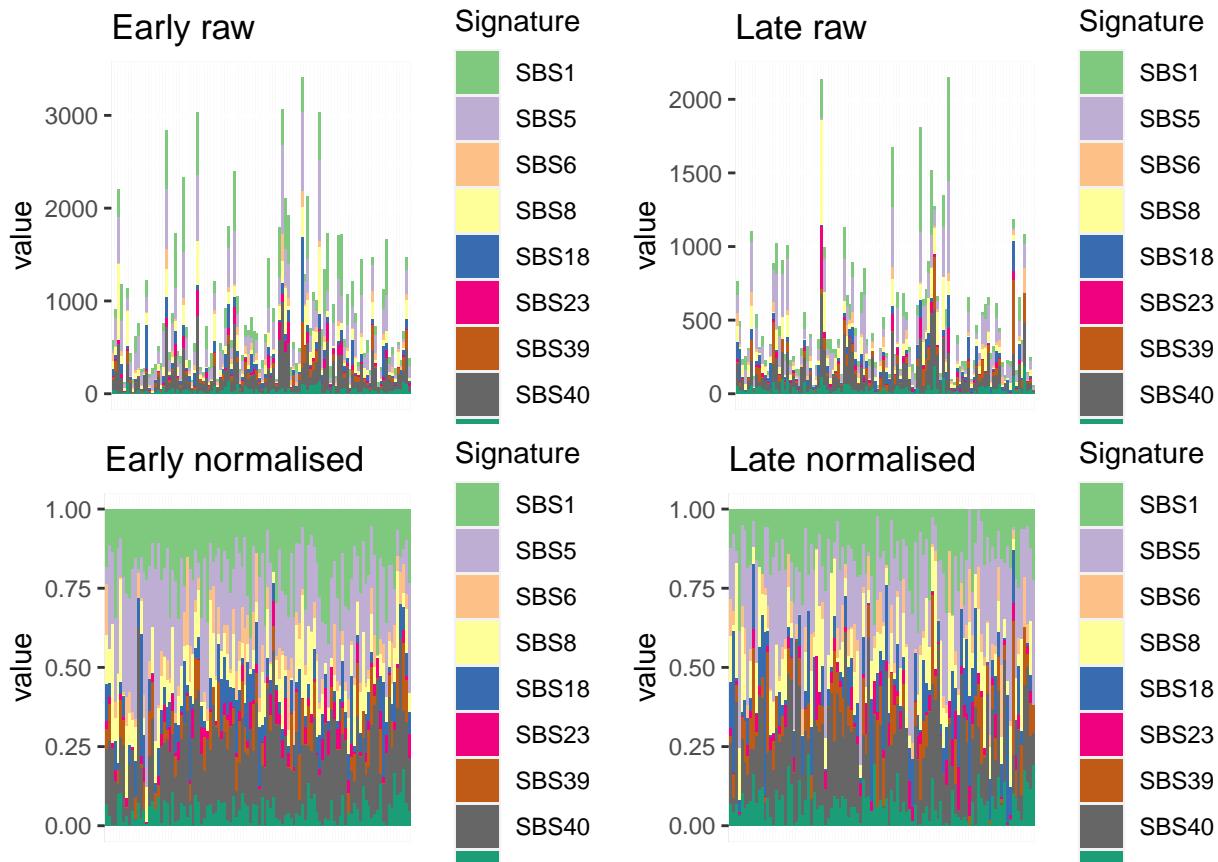
```
## Creating plot... it might take some time if the data are large. Number of samples: 68
```



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

	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)
```

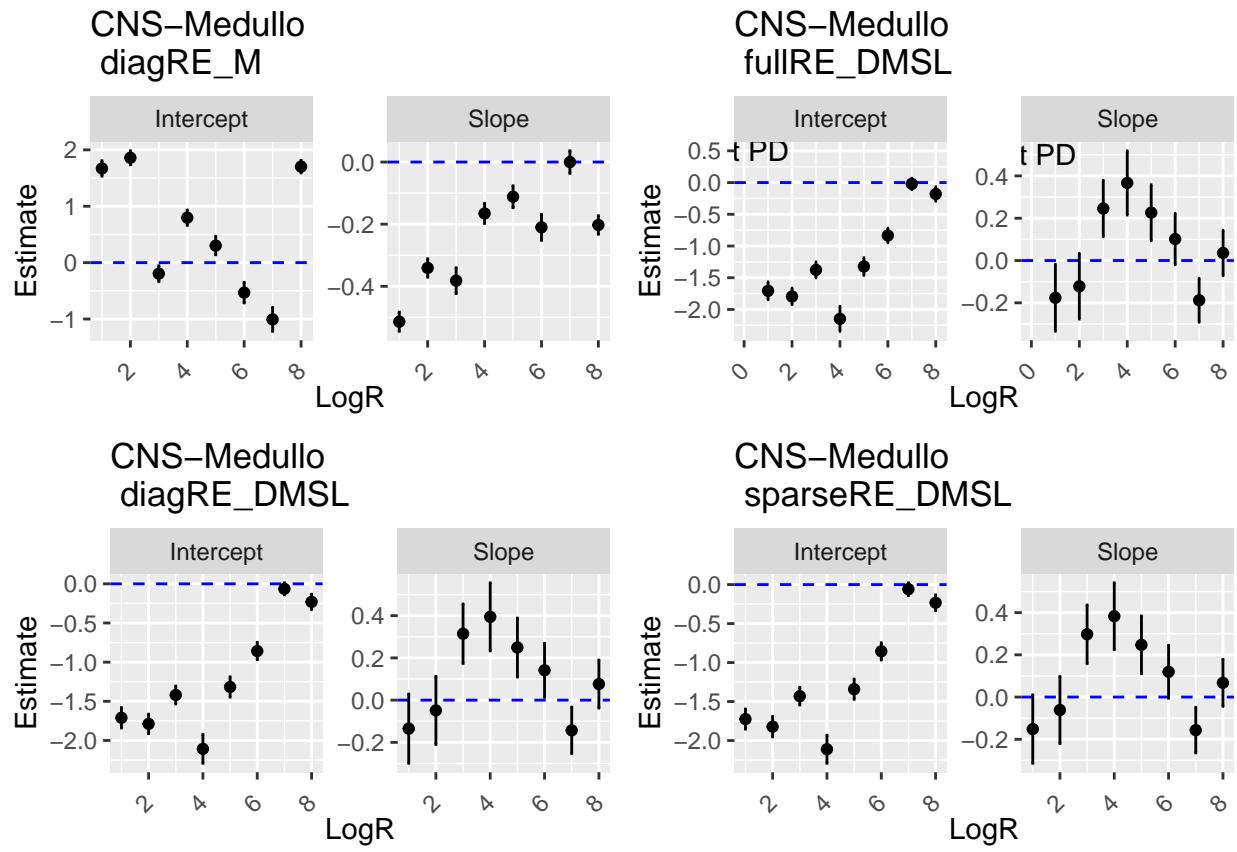
```
##      SBS1      SBS5      SBS6      SBS8      SBS18      SBS23      SBS39
## 0.19177483 0.22946904 0.03737123 0.11614418 0.07466844 0.03836035 0.05498025
##      SBS40      SBS46
## 0.21065558 0.04657610
```

Betas

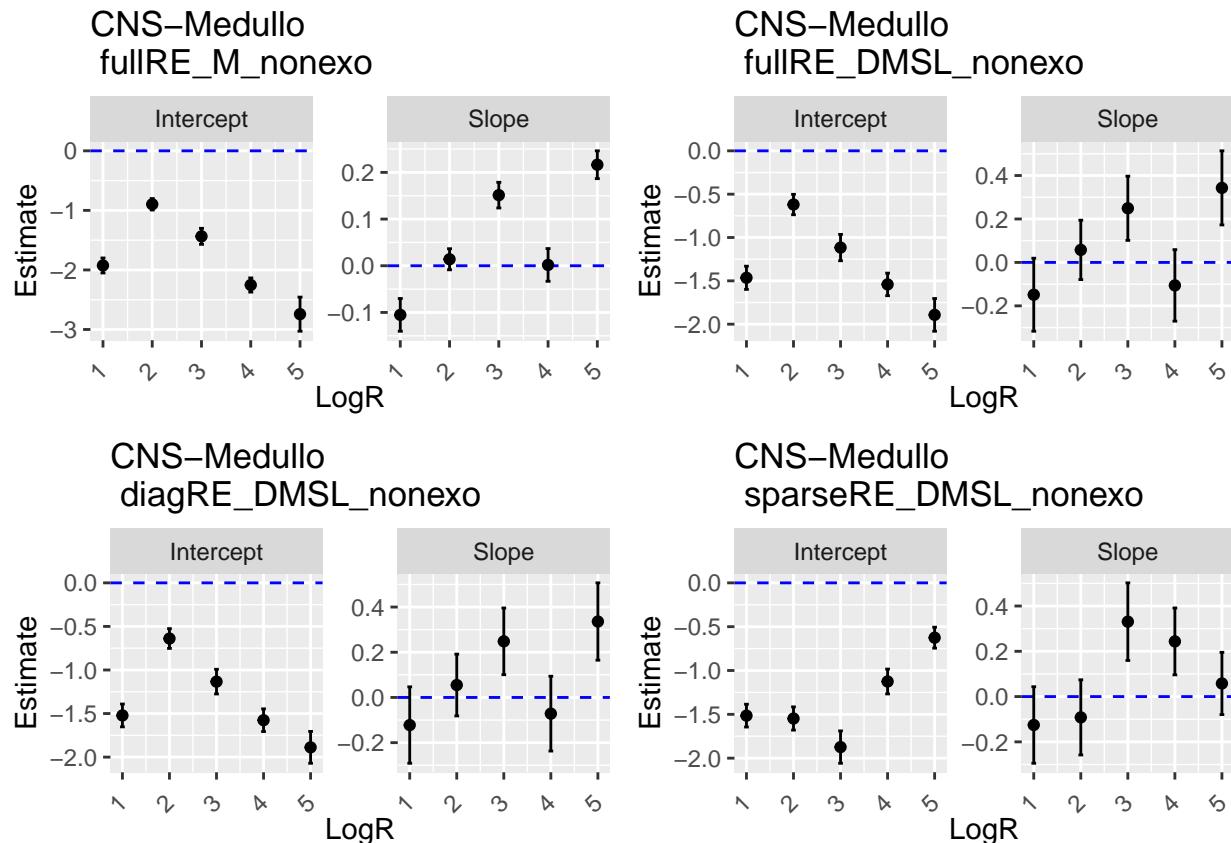
```
ct <- "CNS-Medullo"

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

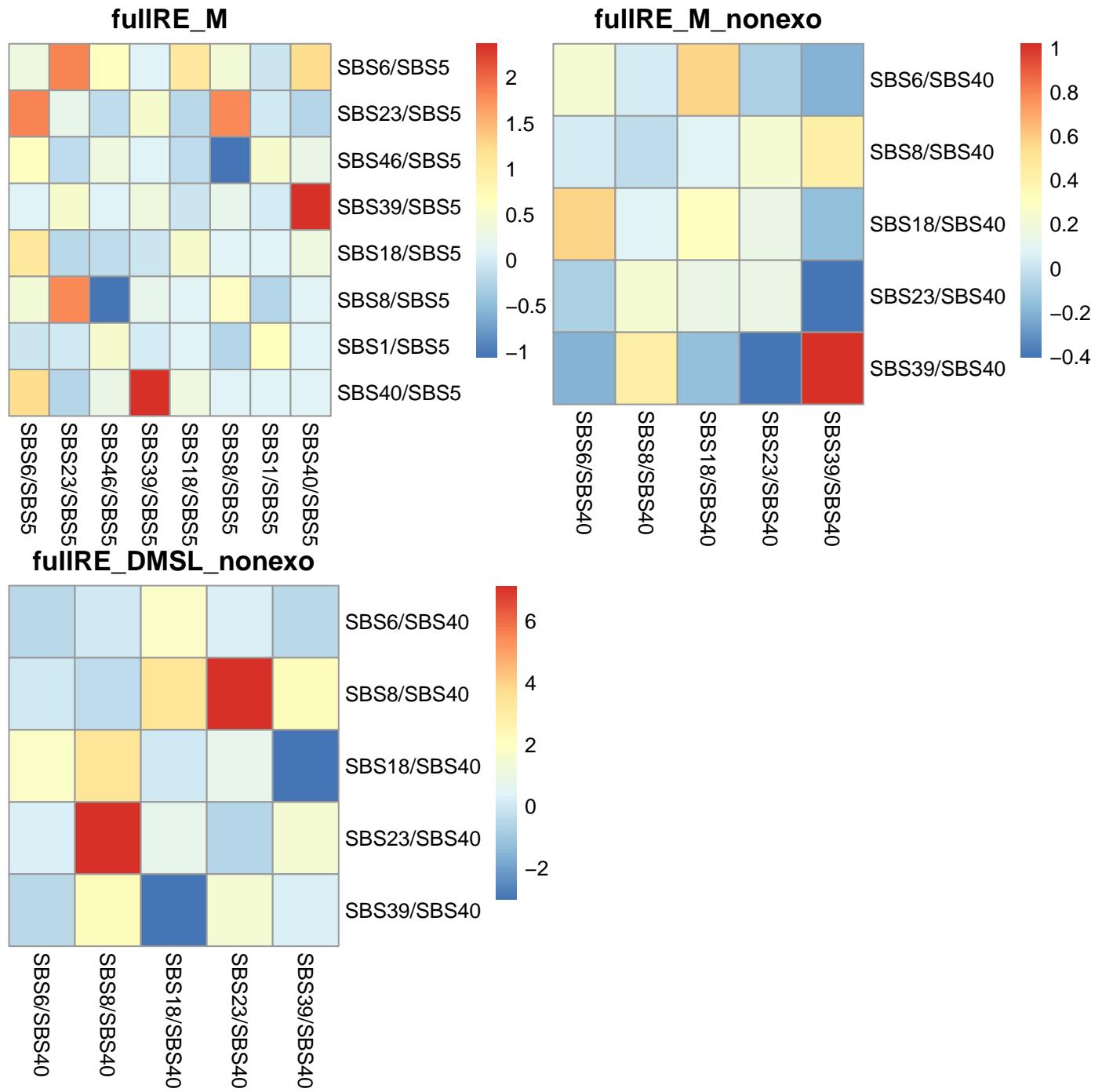
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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.

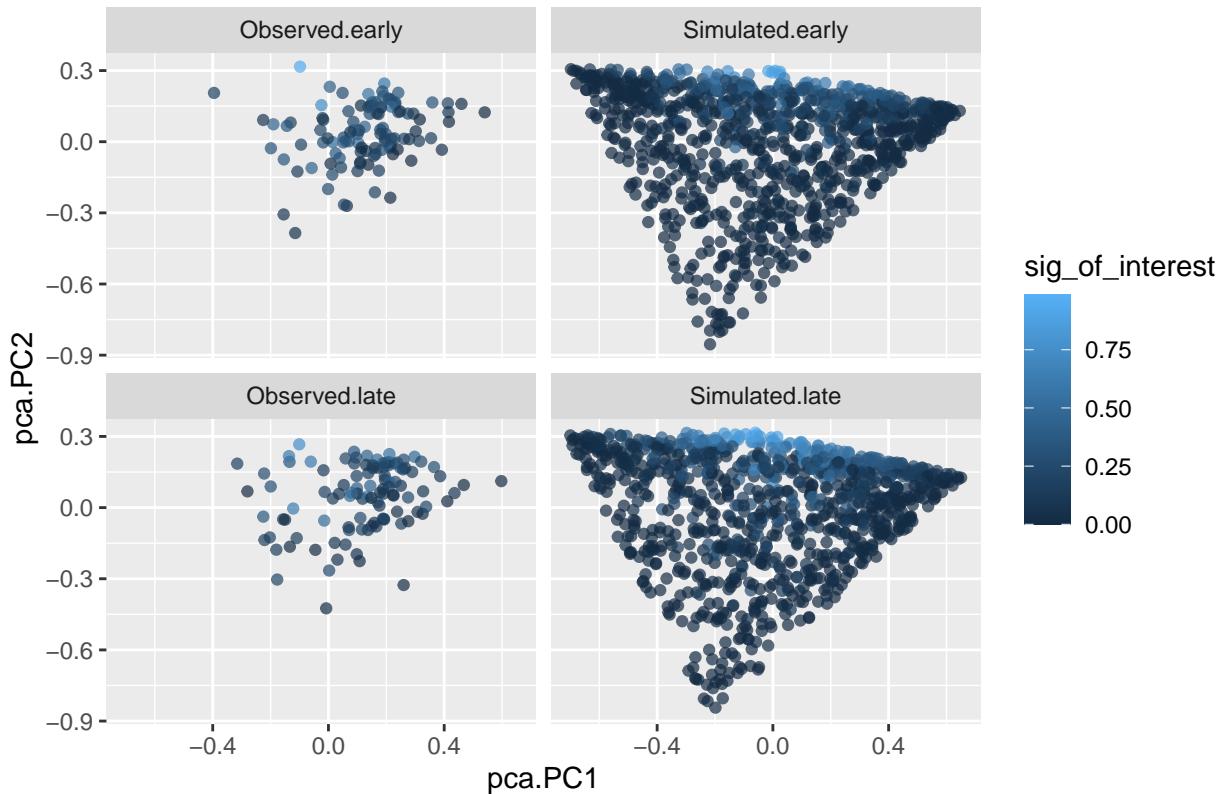
Covariance matrices



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

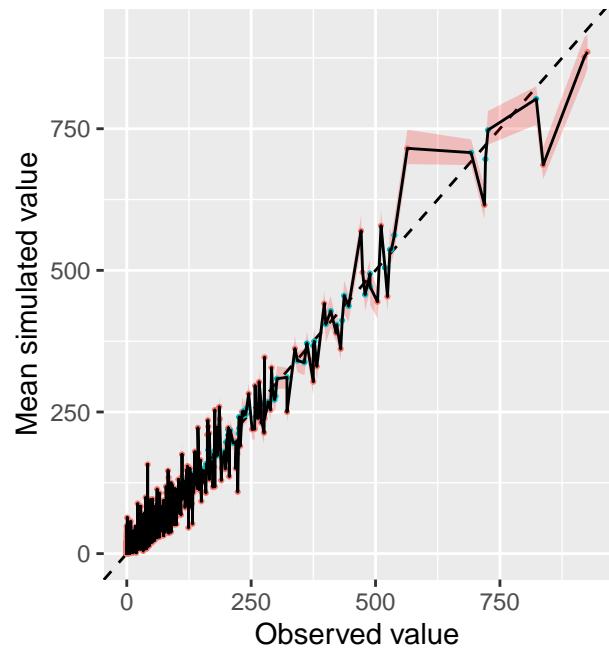


Ranked plot for coverage

```
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)),
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 (DMSL)', ncol=2)
```

obj_CNS_Medullo (M)

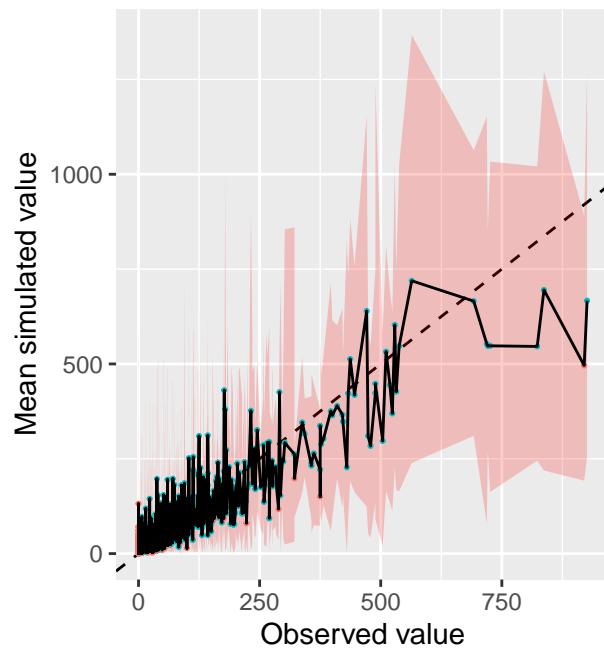
FALSE:812; TRUE:460



col • FALSE • TRUE

obj_CNS_Medullo (DMSL)

FALSE:300; TRUE:972



col • FALSE • TRUE

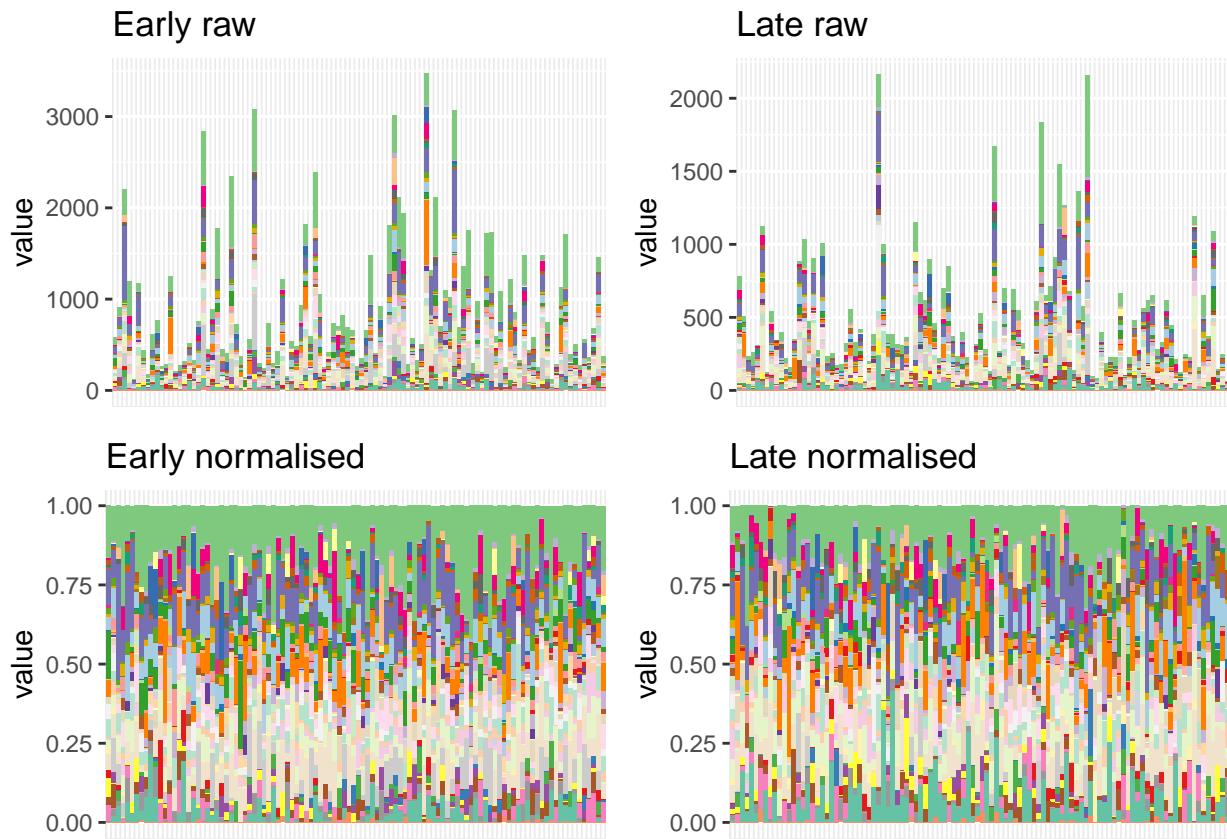
Signatures from mutSigExtractor

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)

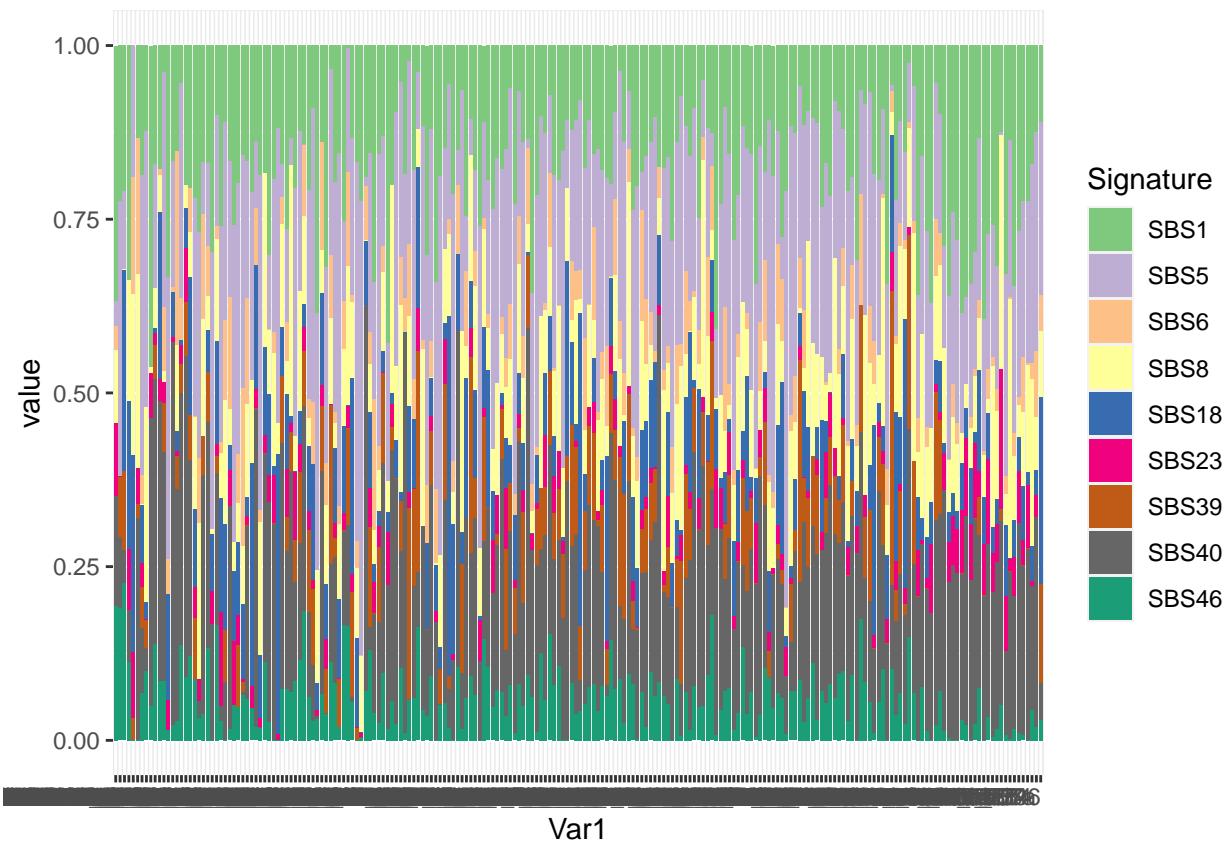
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take some time if the data are large. Number of samples: 106
## Creating plot... it might take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

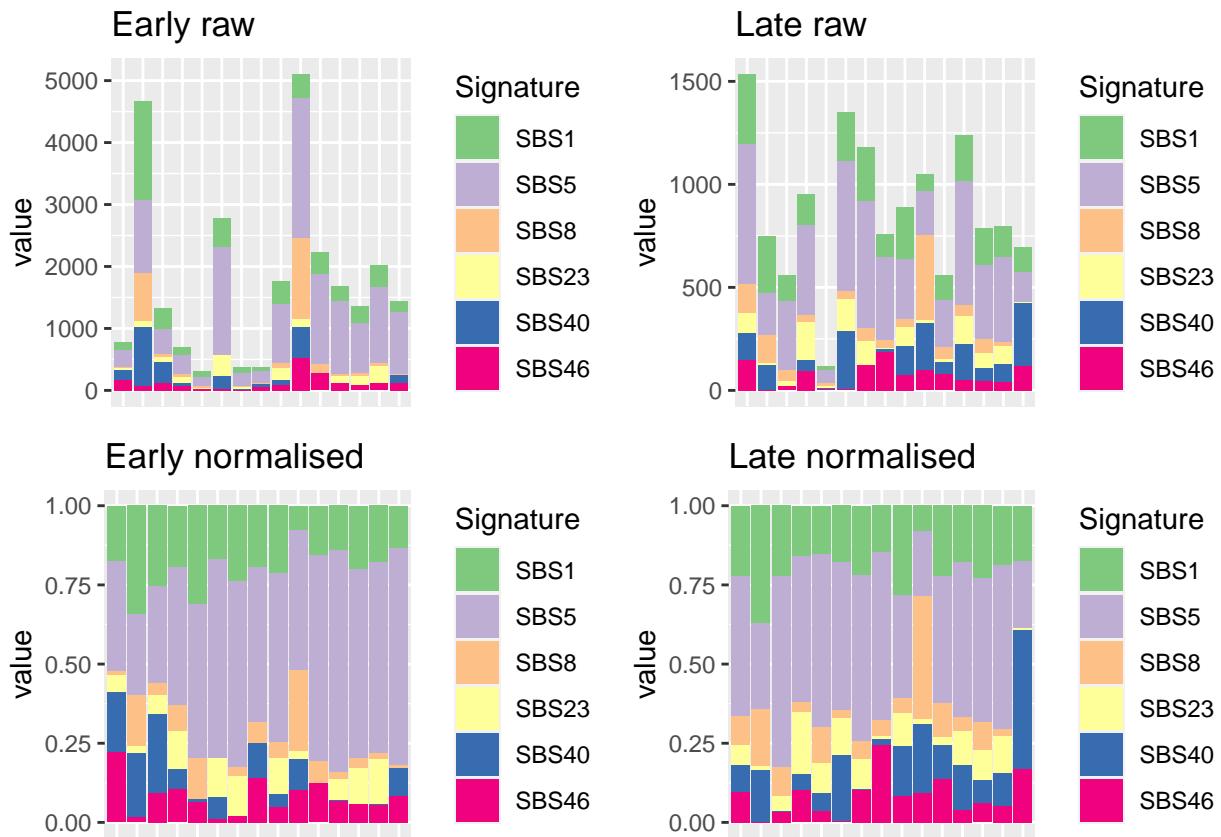
```
## Creating plot... it might take some time if the data are large. Number of samples: 212
```



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

Pretty much everything has converged

		L2	L1
##	value		
## 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

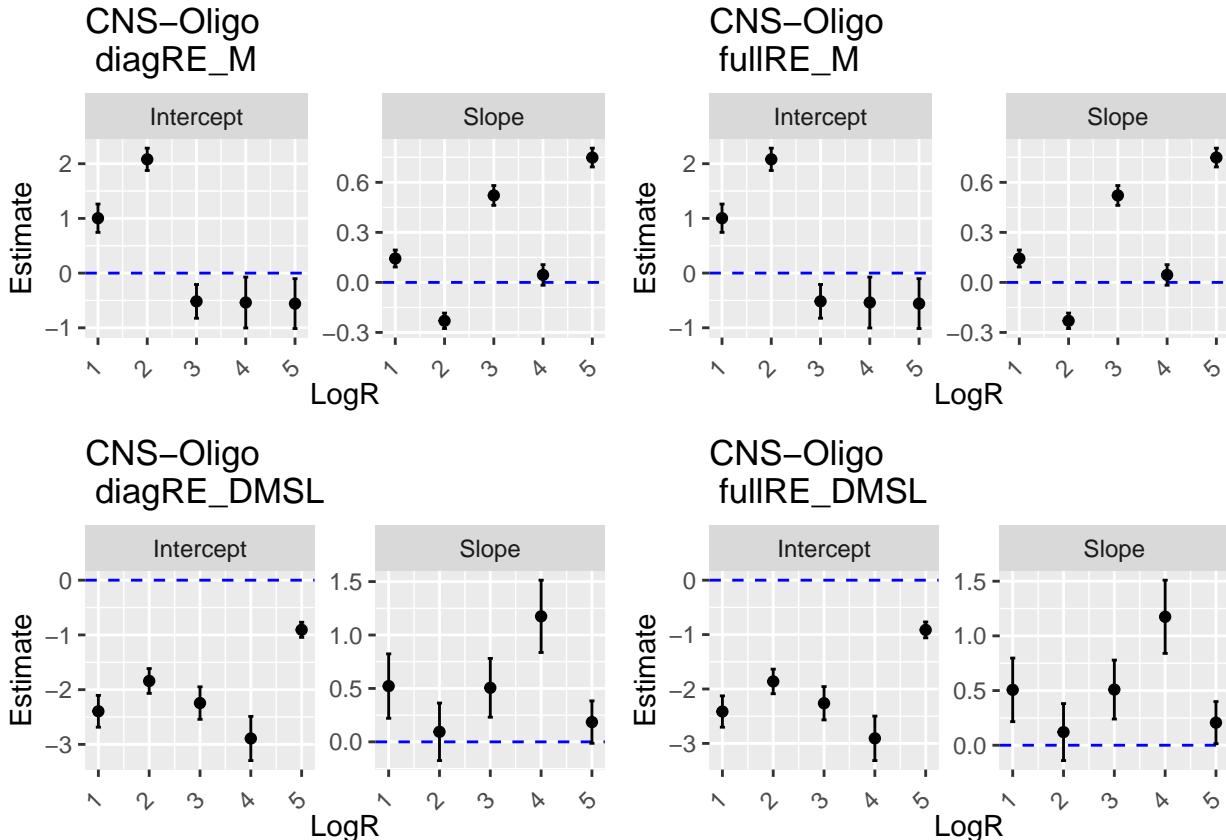
```
## 15 CNS-Oligo
```

```
Timeout fullRE_DMDL_sortednonexo
```

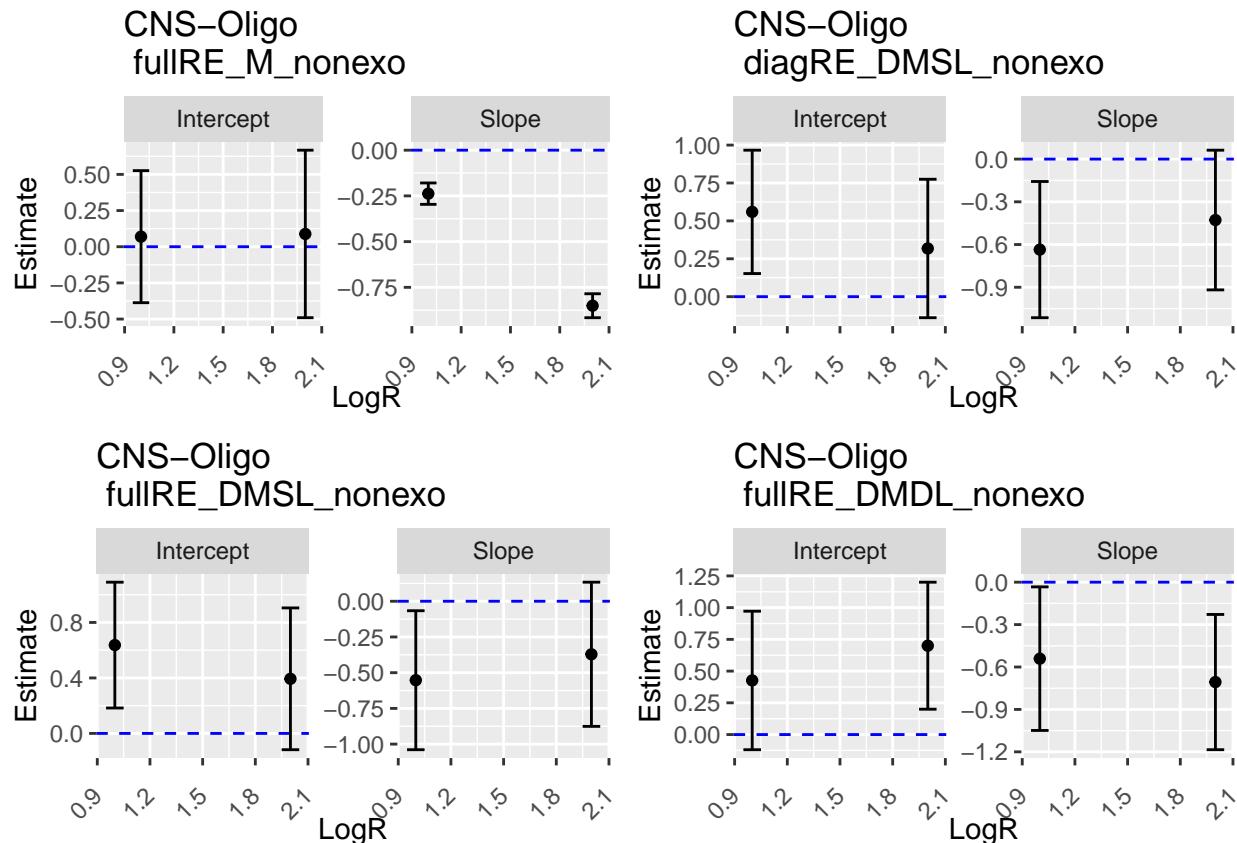
Betas

```
ct <- "CNS-Oligo"
```

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



```
grid.arrange(  
  plot_betas(fullRE_M_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_M_nonexo')),  
  plot_betas(diagRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL_nonexo')),  
  plot_betas(fullRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL_nonexo')),  
  plot_betas(fullRE_DMDL_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMDL_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

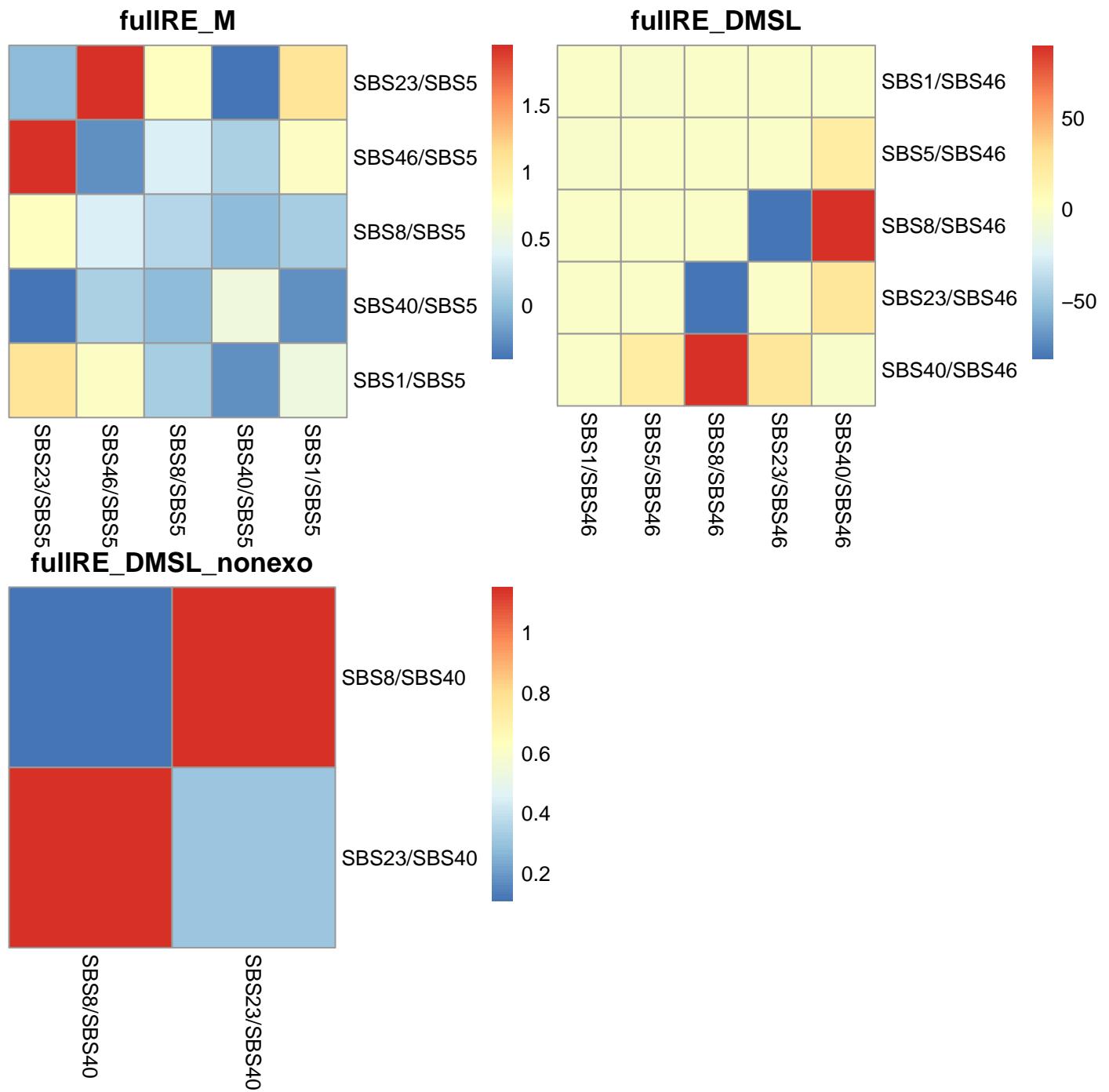
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 0.5220955.

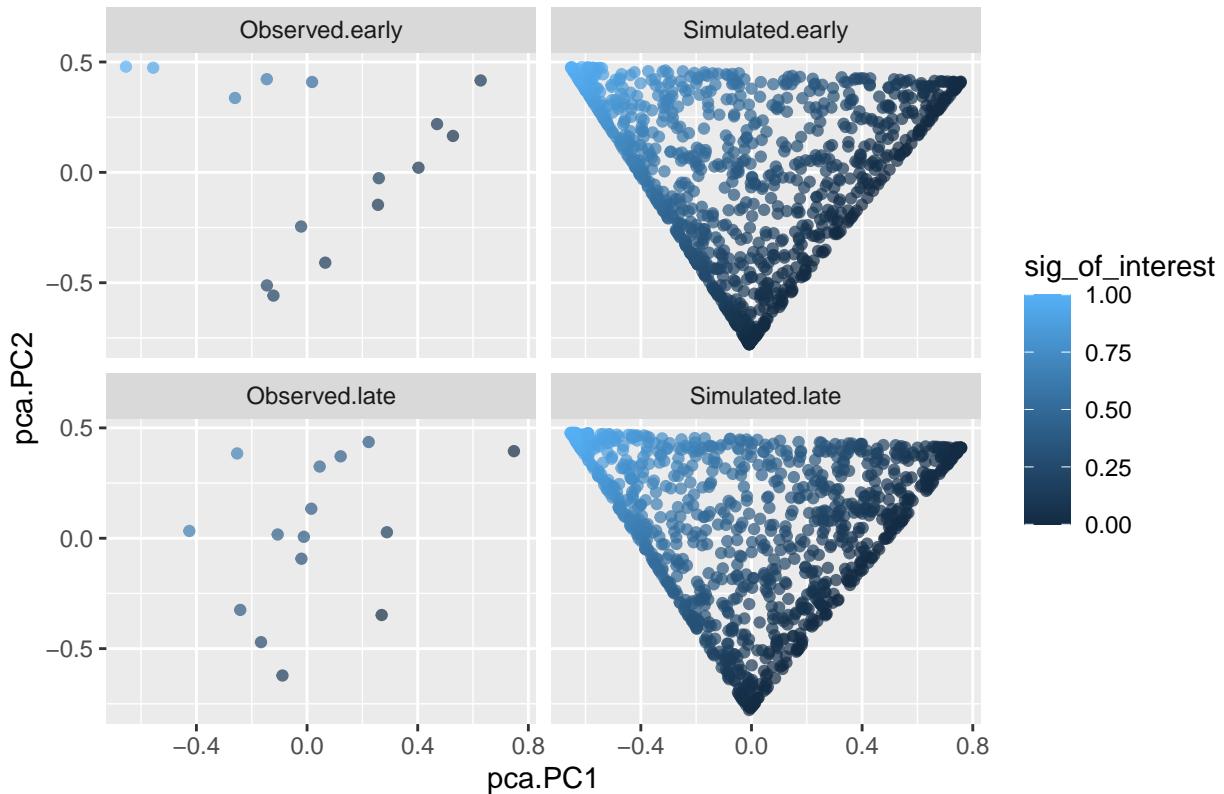
Covariance matrices



Simulation under inferred data

```
## [1] 15
```

Simulation of CNS–Oligo samples

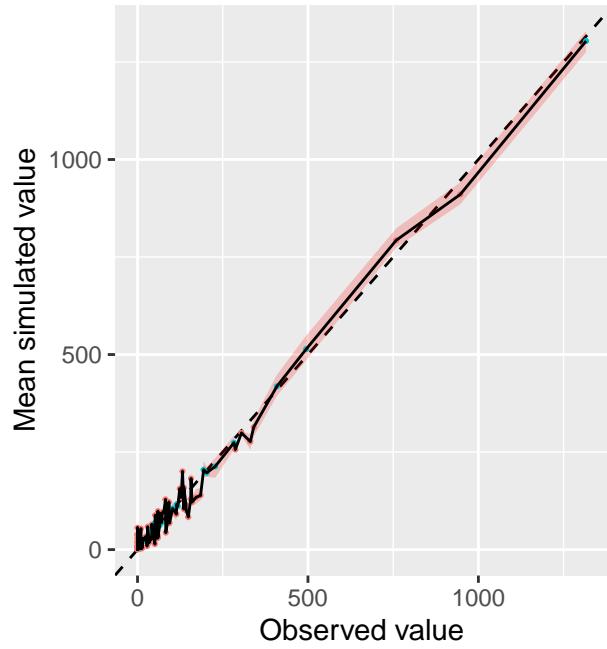


Ranked plot for coverage

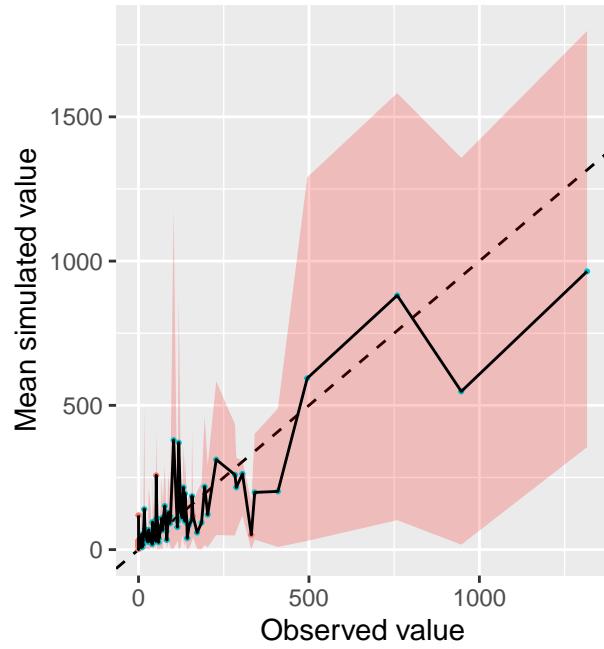
The values for DMSL nonexo look considerably better than for M nonexo.

```
ct <- "CNS-Oligo"
integer_overdispersion_param_DMSL <- 1
obj_CNS_Oligo_nonexo <- give_subset_sigs_TMBobj(obj_CNS_Oligo, 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_Oligo_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_Oligo_nonexo,
loglog = F, title = 'obj_CNS_Oligo (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_CNS_Oligo_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_CNS_Oligo_nonexo,
loglog = F, title = 'obj_CNS_Oligo (DMSL)', ncol=2)
```

obj_CNS_Oligo (M)
FALSE:68; TRUE:22



obj_CNS_Oligo (DMSL)
FALSE:17; TRUE:73



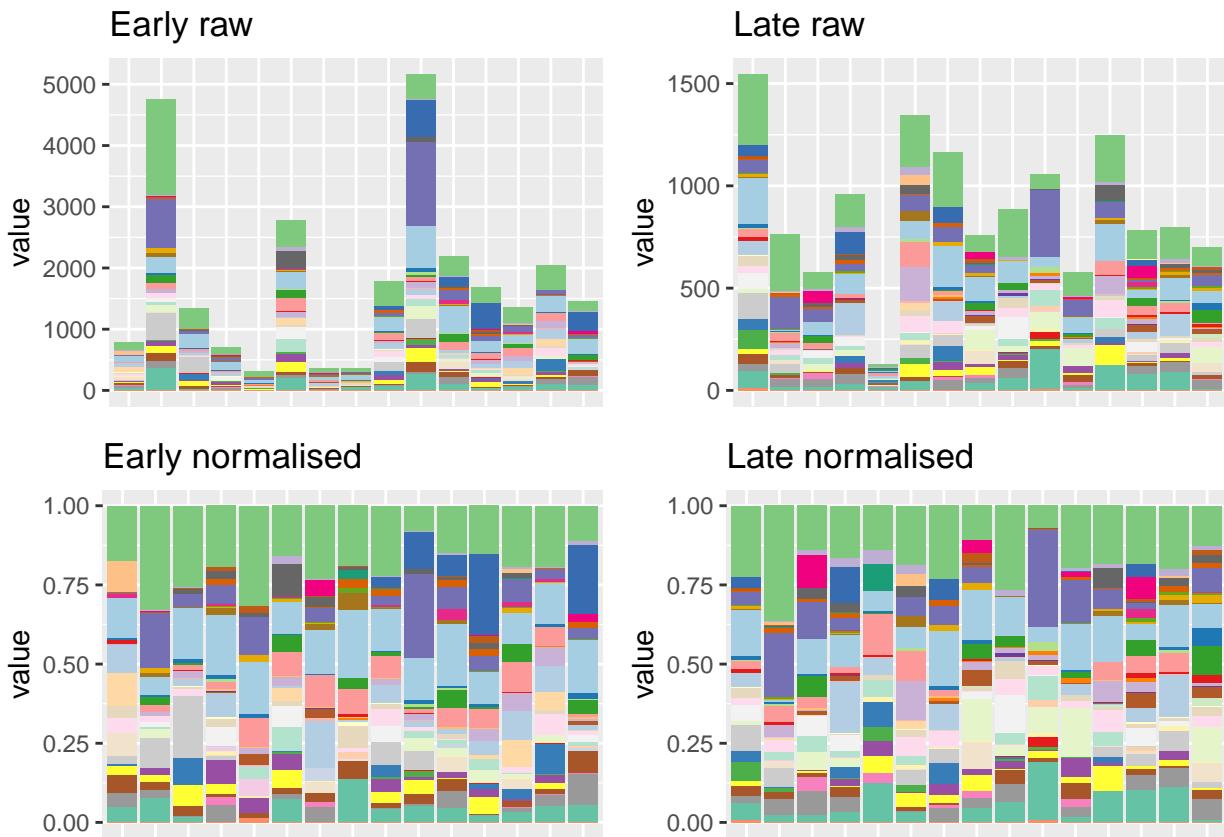
Signatures from mutSigExtractor

These are the signatures from mutSigExtractor:

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

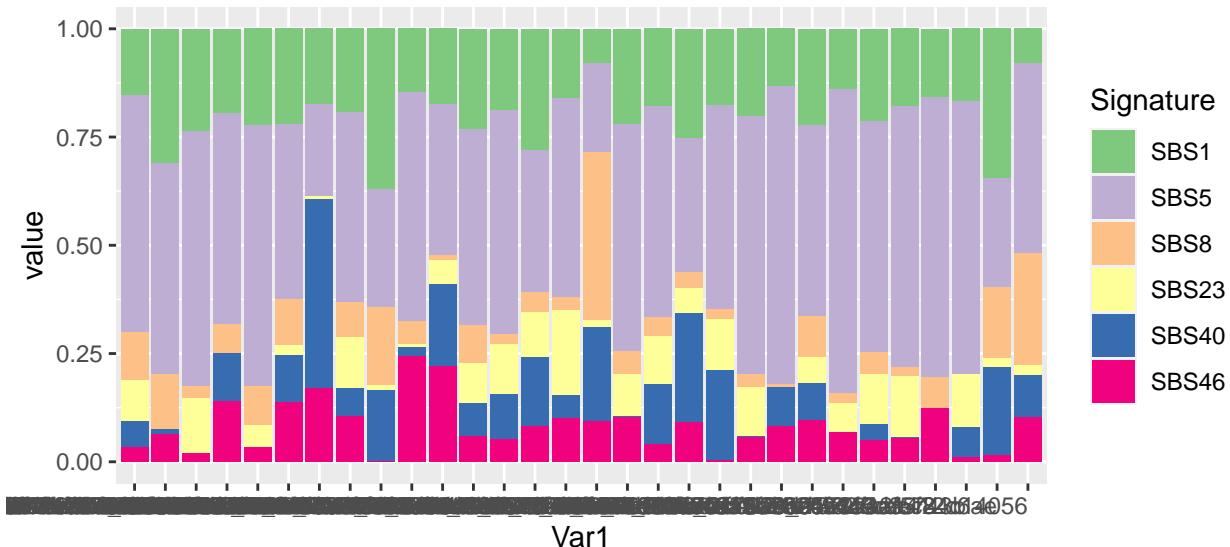
## [1] 15
give_barplot_from_obj(obj = obj_CNS_Oligo_mutSigExtractor, legend_on = FALSE)

## 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
```



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

Creating plot... it might take some time if the data are large. Number of samples: 30



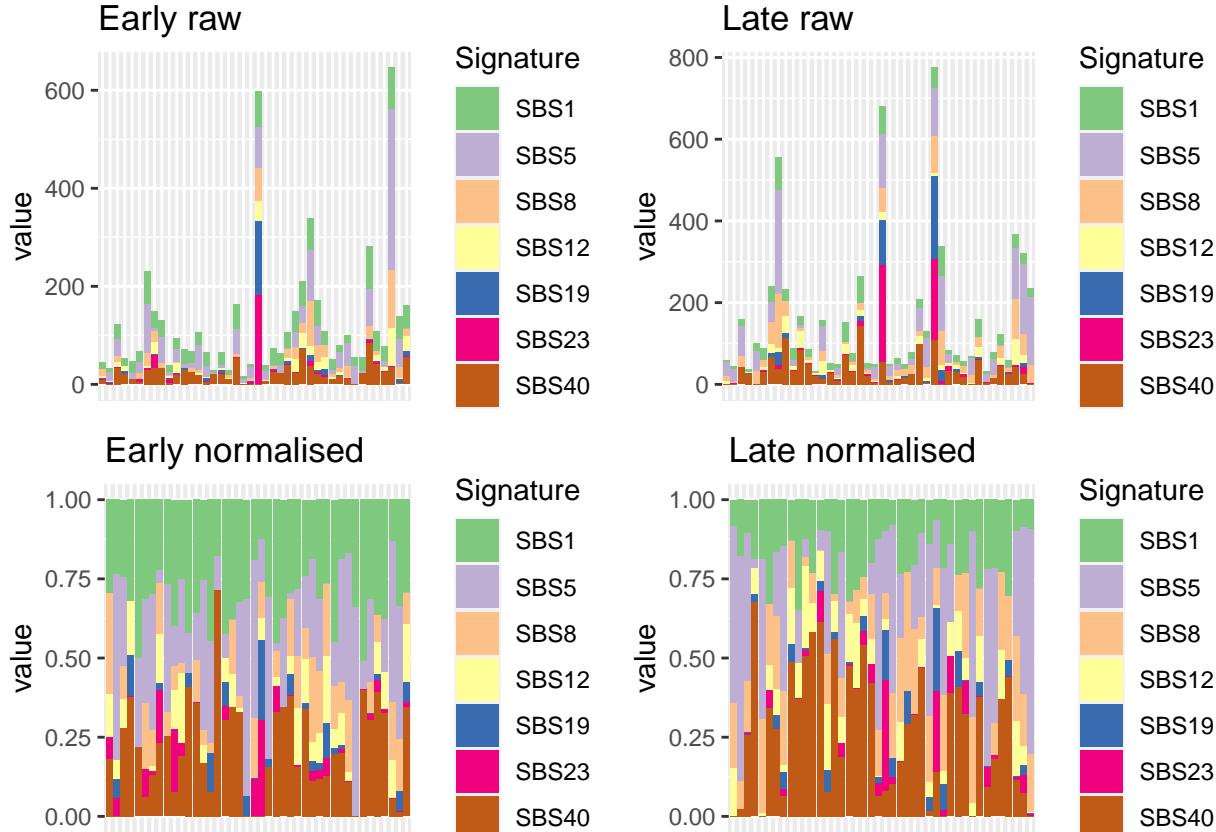
CNS-PiloAstro

CNS-PiloAstro

Barplot and general statistics

```
## [1] 42
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 42
## Creating plot... it might take some time if the data are large. Number of samples: 42
## Creating plot... it might take some time if the data are large. Number of samples: 42
## Creating plot... it might take some time if the data are large. Number of samples: 42
```



The number of samples and signatures is:

```
## [1] 84 7
```

The signatures are:

```
## [1] "SBS1"  "SBS5"  "SBS8"  "SBS12" "SBS19" "SBS23" "SBS40"
```

Convergence table

We have converged results for everything except for full RE DM, in the case of all signatures (with only nonexo everything has).

```
##          value           L2           L1
## 1 CNS-PiloAstro hessian_positivedefinite_bool diagRE_M
## 2 CNS-PiloAstro hessian_positivedefinite_bool fullRE_M
```

```

## 3 CNS-PiloAstro hessian_positivedefinite_bool diagRE_DMDL
## 4 CNS-PiloAstro hessian_nonpositivedefinite_bool fullRE_halfDM
## 5 CNS-PiloAstro hessian_nonpositivedefinite_bool fullRE_DMDL
## 6 CNS-PiloAstro hessian_positivedefinite_bool diagRE_DMSL
## 7 CNS-PiloAstro hessian_positivedefinite_bool sparseRE_DMSL
## 8 CNS-PiloAstro hessian_nonpositivedefinite_bool fullRE_DMSL
## 9 CNS-PiloAstro hessian_nonpositivedefinite_bool fullRE_DMSL_SBS1
## 10 CNS-PiloAstro hessian_positivedefinite_bool fullRE_M_nonexo
## 11 CNS-PiloAstro hessian_positivedefinite_bool diagRE_DMSL_nonexo
## 12 CNS-PiloAstro hessian_positivedefinite_bool sparseRE_DMSL_nonexo
## 13 CNS-PiloAstro hessian_positivedefinite_bool fullRE_DMSL_nonexo
## 14 CNS-PiloAstro hessian_nonpositivedefinite_bool fullRE_DMDL_nonexo
## 15 CNS-PiloAstro hessian_positivedefinite_bool fullRE_DMDL_sortednonexo

```

Re-running of fitting

Using fullRE_M to fit fullRE_DMSL (all sigs, as the one with nonexo has already converged)

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

If we use the values of the fullRE M as initial values for the fullRE DMSL still do not converge:

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

Potentially problematic signatures

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

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

```
#> ##      SBS1      SBS5      SBS8      SBS12      SBS19      SBS23      SBS40
#> ## 0.0000000 0.2261905 0.2023810 0.2857143 0.5119048 0.5000000 0.1190476
```

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

```
#> ##      SBS1      SBS5      SBS8      SBS12      SBS19      SBS23      SBS40
#> ## 0.19840611 0.26357297 0.14212187 0.07313631 0.05894073 0.06749128 0.19633073
```

SBS19 and SBS23 are quite sparse.

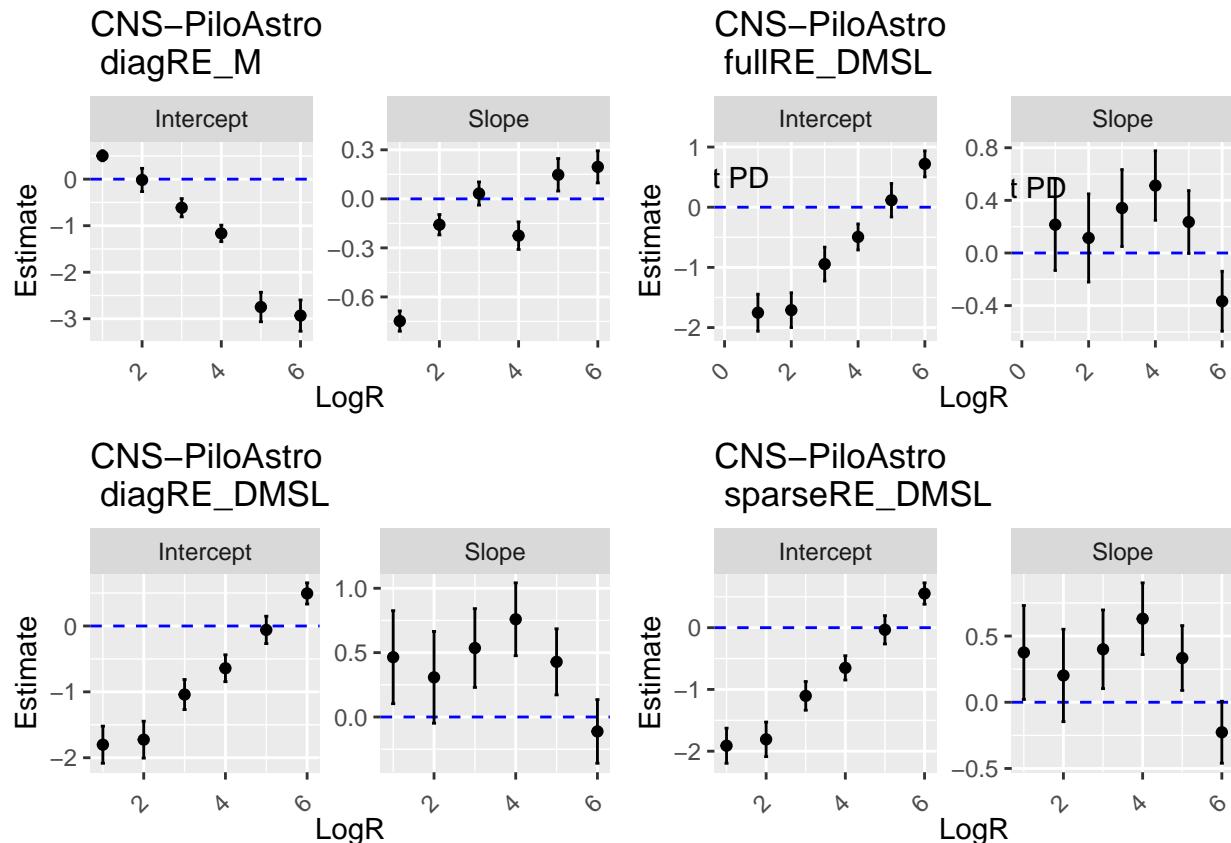
```
additional_sortedMnonexo[["CNS-PiloAstro"]] <- sortedM_CNSPiloAstro
additional_sortedDMSLnonexo[["CNS-PiloAstro"]] <- sortedDM_CNSPiloAstro
```

Betas

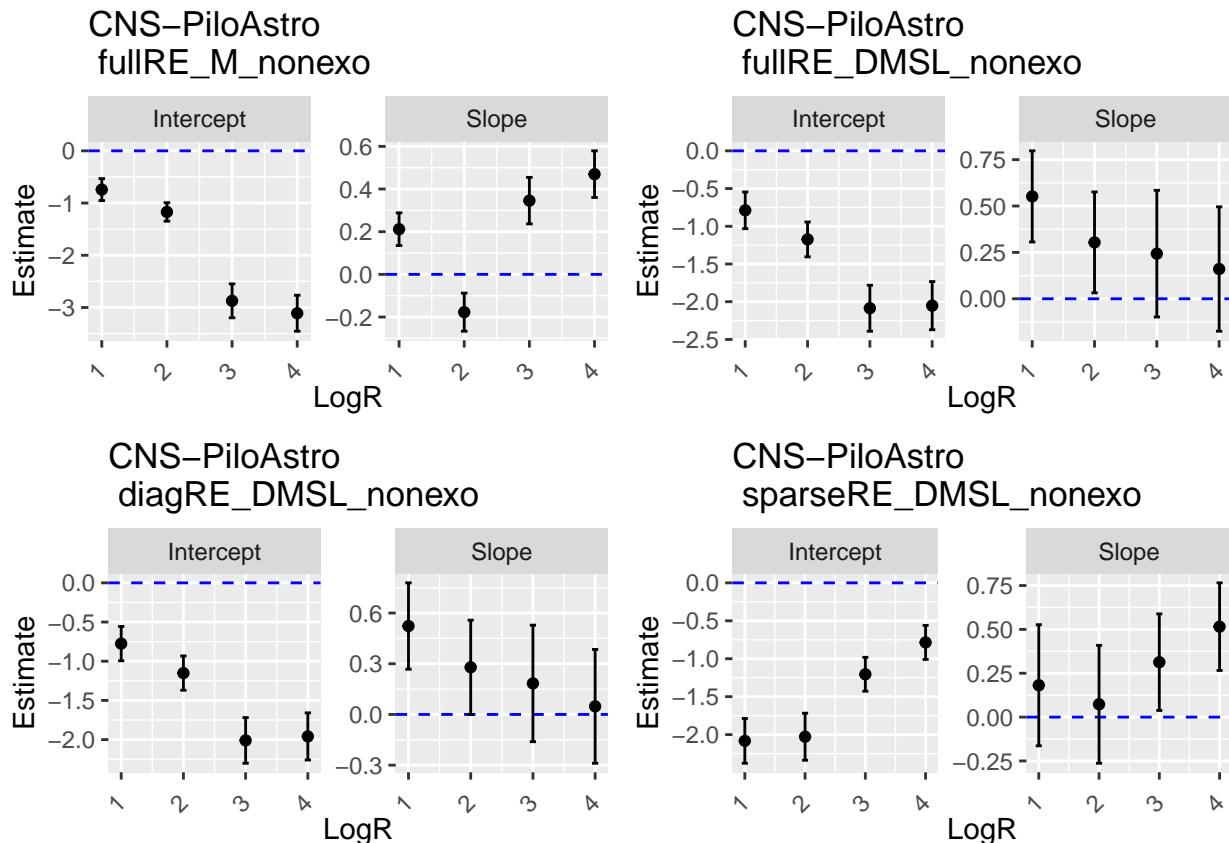
```
ct <- "CNS-PiloAstro"

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

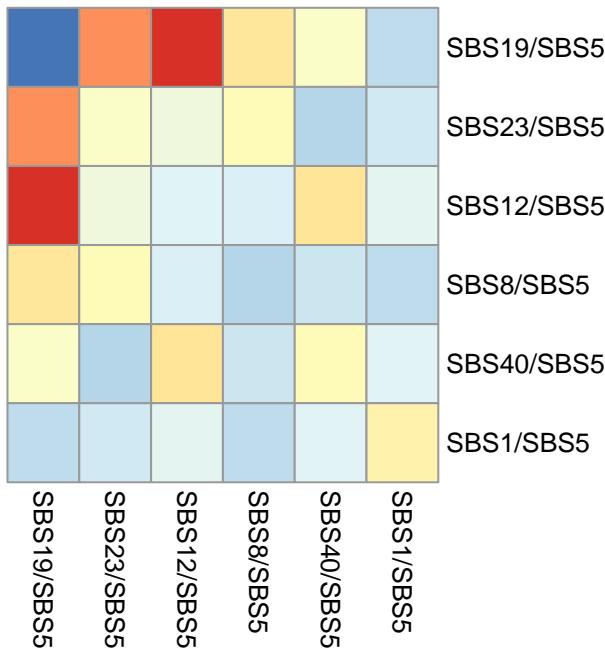
## Warning in wald_generalised(v = i$par.fixed[idx_beta], sigma =
## i$cov.fixed[idx_beta, : 20201218: sigma** (1/2) has now been replaced by (as we
## had before sometime in November) sigma

```

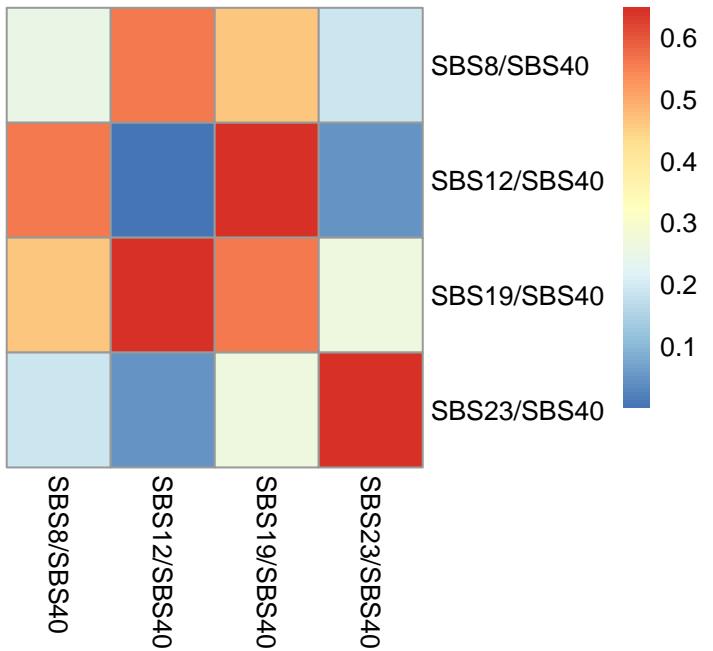
We use the results from the full RE single lambda DM to test for differential abundance, giving a p-value of 0.2632004.

Covariance matrices

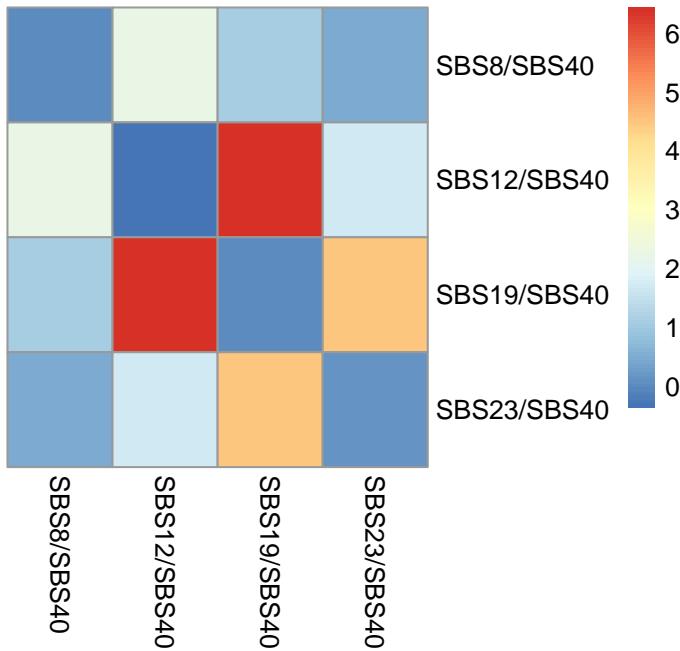
fullRE_M



fullRE_M_nonexo



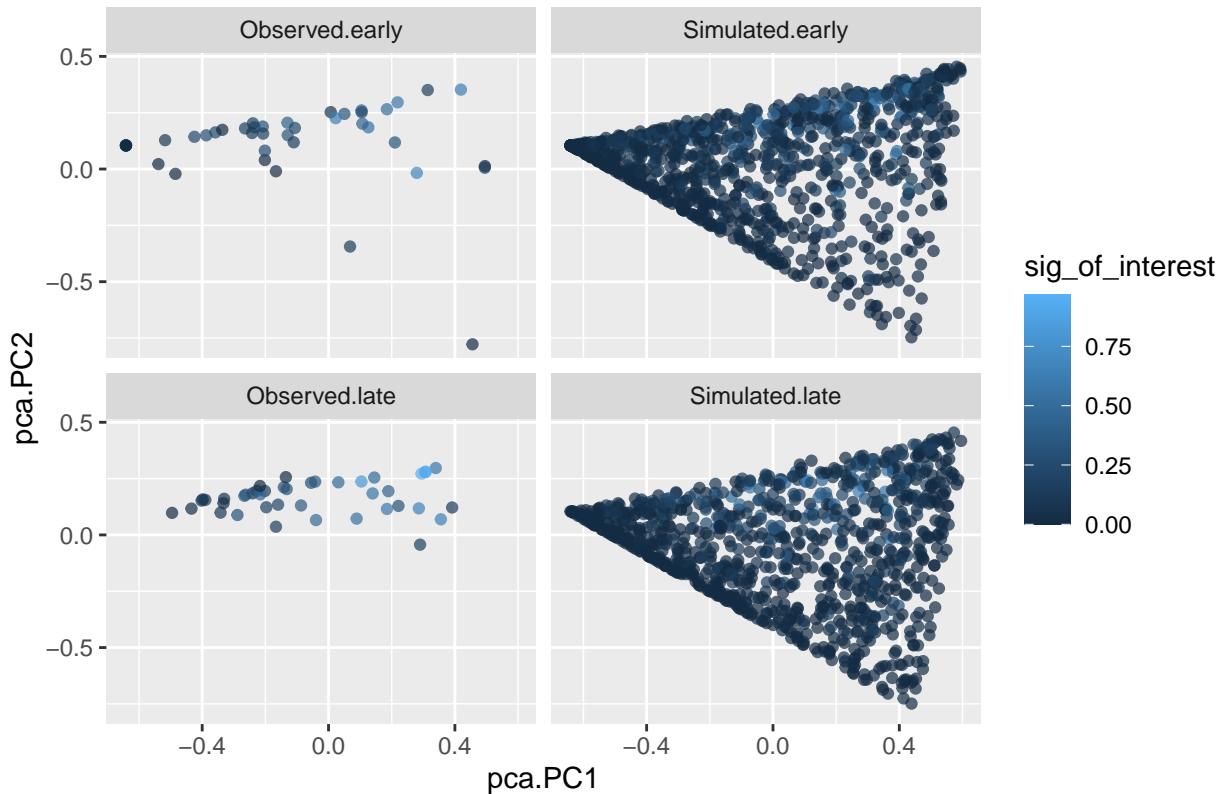
fullRE_DMSL_nonexo



Simulation under inferred data

```
## [1] 42
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

Simulation of CNS–PiloAstro samples

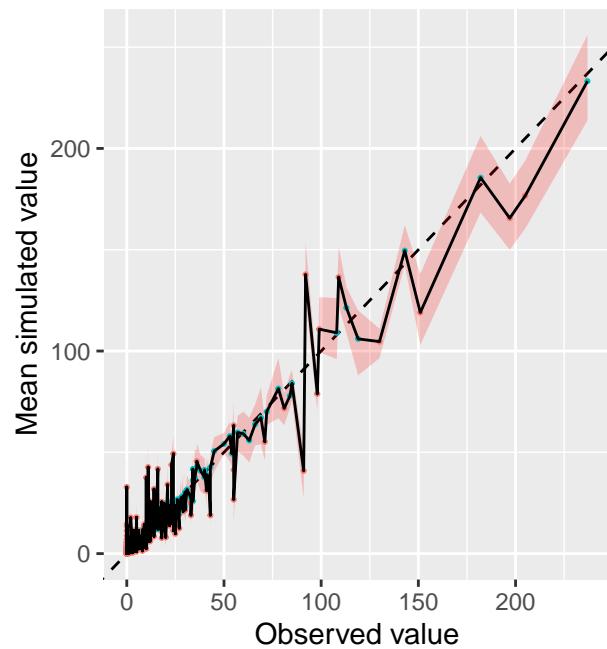


Ranked plot for coverage

```
ct <- "CNS-PiloAstro"
integer_overdispersion_param_DMSL <- 1
obj_CNS_PiloAstro_nonexo <- give_subset_sigs_TMBobj(obj_CNS_PiloAstro, 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_PiloAstro_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_PiloAstro_nonexo,
loglog = F, title = 'obj_CNS_PiloAstro (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_CNS_PiloAstro_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_CNS_PiloAstro_nonexo,
loglog = F, title = 'obj_CNS_PiloAstro (DMSL)', ncol=2)
```

obj_CNS_PiloAstro (M)

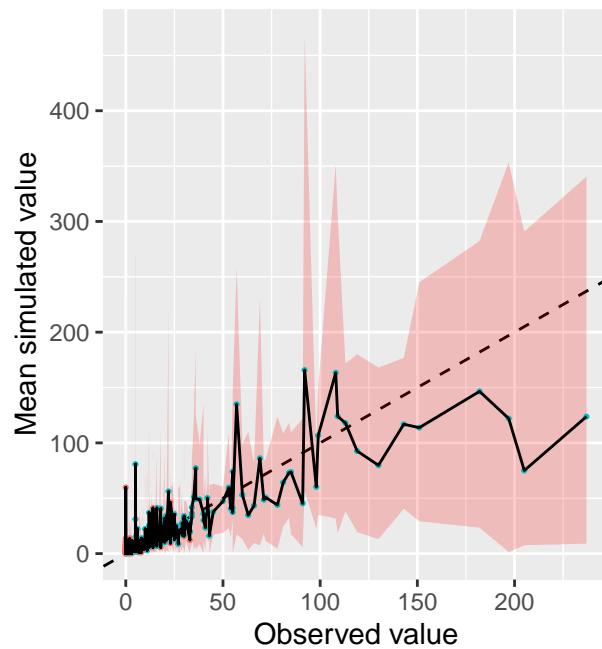
FALSE:249; TRUE:166



col ● FALSE ● TRUE

obj_CNS_PiloAstro (DMSL)

FALSE:155; TRUE:260



col ● FALSE ● TRUE

Signatures from mutSigExtractor

The signatures from mutSigExtractor are a bit more chaotic:

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

## [1] 42

give_barplot_from_obj(obj = obj_CNS_PiloAstro_mutSigExtractor, legend_on = FALSE)

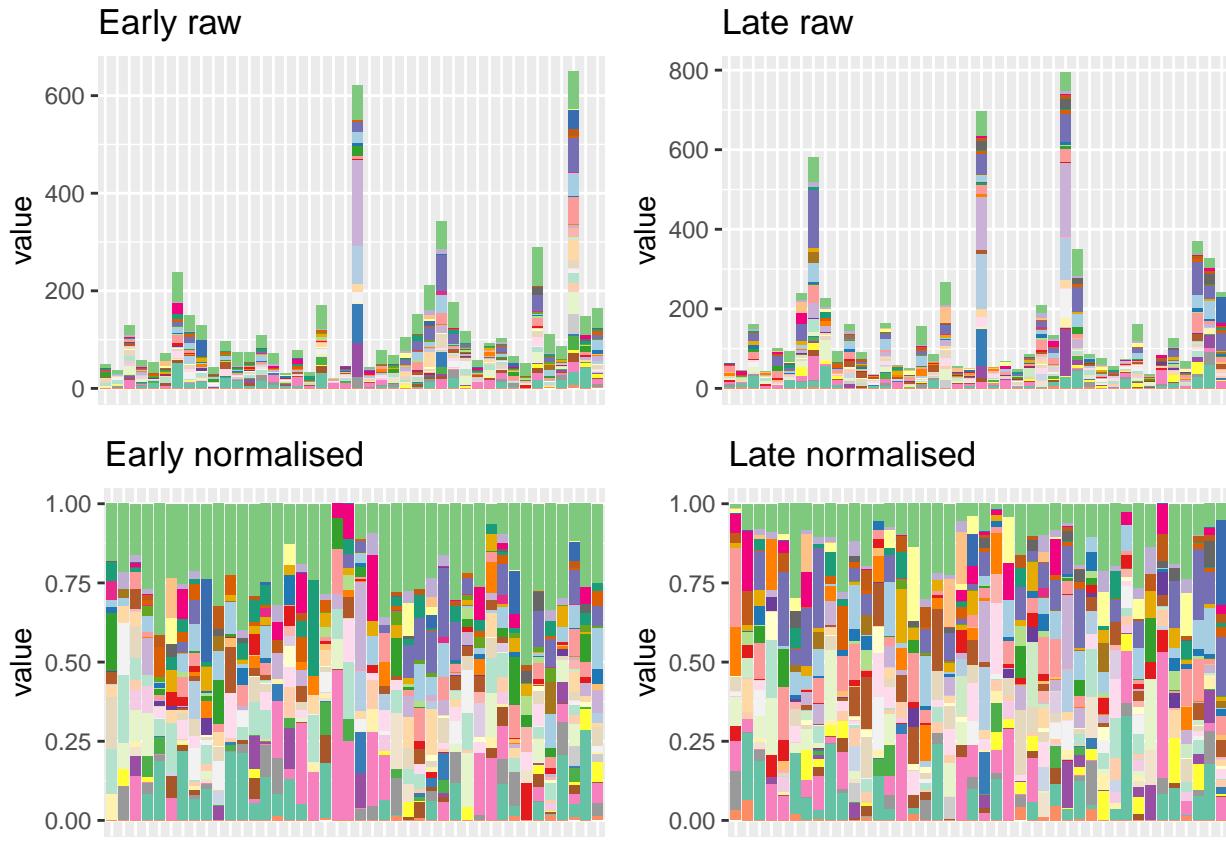
## Creating plot... it might take some time if the data are large. Number of samples: 42
## Creating plot... it might take some time if the data are large. Number of samples: 42
## Creating plot... it might take some time if the data are large. Number of samples: 42
## Creating plot... it might take some time if the data are large. Number of samples: 42

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

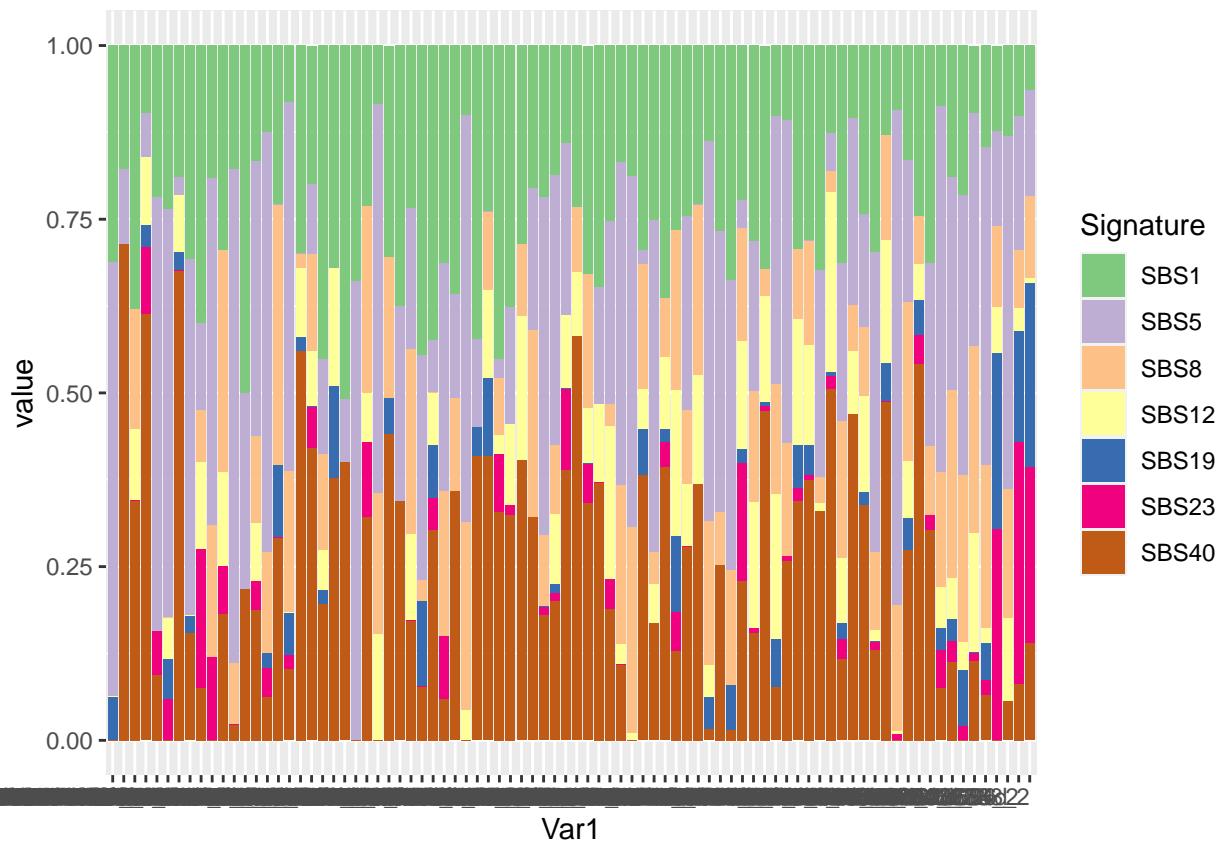
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



Exposures sorted by increasing number of mutations: there is no trend of signatures being associated with the number of mutations, with perhaps SBS9 being slightly found in the rightmost side preferentially.

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

```
## Creating plot... it might take some time if the data are large. Number of samples: 84
```



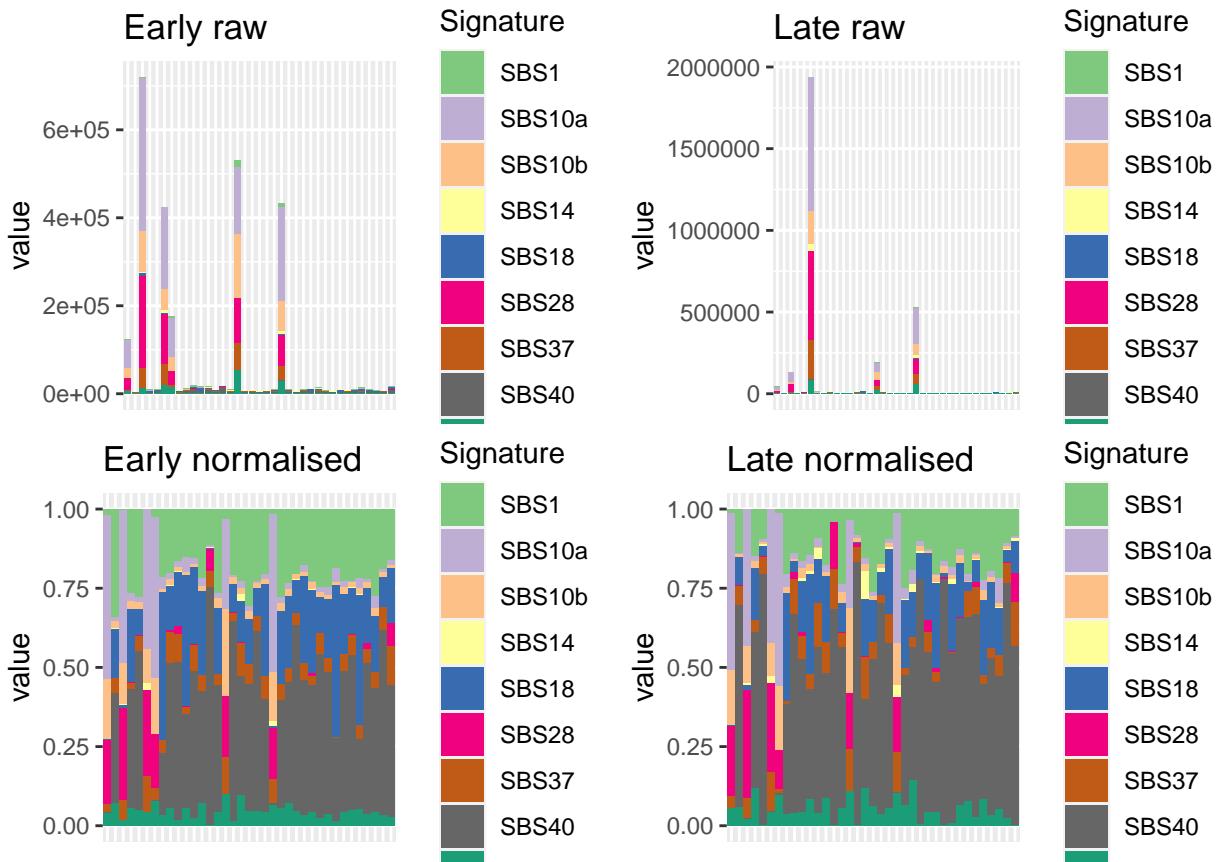
ColoRect-AdenoCA

ColoRect-AdenoCA

Barplot and general statistics

```
## [1] 37
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 37
## Creating plot... it might take some time if the data are large. Number of samples: 37
## Creating plot... it might take some time if the data are large. Number of samples: 37
## Creating plot... it might take some time if the data are large. Number of samples: 37
```



The number of samples and signatures is:

```
## [1] 74 9
```

The signatures are:

```
## [1] "SBS1"   "SBS10a" "SBS10b" "SBS14"   "SBS18"   "SBS28"   "SBS37"   "SBS40"
## [9] "SBS44"
```

Convergence table

We only have converged results for the multinomial with diag RE, when including all mutations. For exogenous mutations, full DMSL is has not converged.

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

```

## 12 ColoRect-AdenoCA    hessian_positivedefinite_bool      sparseRE_DMSL_nonexo
## 13 ColoRect-AdenoCA hessian_nonpositivedefinite_bool      fullRE_DMSL_nonexo
## 14 ColoRect-AdenoCA hessian_nonpositivedefinite_bool      fullRE_DMDL_nonexo
## 15 ColoRect-AdenoCA                           Timeout fullRE_DMDL_sortednonexo

```

Re-running of fitting

Using fullRE_M_nonexo to fit fullRE_DMSL_nonexo

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

Now the fullM doesn't converge (even though the original fullRE M nonexo did converge?), so I cannot use all the parameters to find the starting parameters of the DM, as some are NA. I can however use some, such as beta.

What parameters are NA?

```

##          beta          beta          beta          beta          beta          beta          beta
## -2.54404819  0.59706447  0.63355077 -0.46879338  4.20343871  0.18574156
##          beta          beta          beta          beta          beta          beta          beta
## -19.99432403  0.29636775  3.02975360  0.28475722 -1.60129347  0.01913378
##          beta          beta cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE
##  5.93971517  0.15499188 -12.04185894 -16.74375580  7.37447864  9.17341782
## cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE
## -17.09070185 -8.65885911  6.41420768 -10.90041616 -20.09373728  4.61570780
## cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE
##  7.72147141 -12.99908453 -4.22783276  9.21589607 -2.22915894 -0.93447317
## cov_par_RE cov_par_RE cov_par_RE cov_par_RE cov_par_RE logs_sd_RE
## 10.83144466  9.10923370 -7.69300283 -16.65325966 -2.26097772 18.34680278
## logs_sd_RE logs_sd_RE logs_sd_RE logs_sd_RE logs_sd_RE logs_sd_RE
## 10.36960317  9.28298541 27.22237889  5.86088489  3.32760927  5.93365712

```

Betas, logsd and covariances are not NA. Therefore, we use these values as starting values, and give an empty random effects matrix.

I get the error “gradient function must rerurn a number vector of length 43” for some reason I don’t understand - it’s as though the initial values I am giving are not correct.

Potentially problematic signatures

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

```

##      SBS1      SBS10a     SBS10b      SBS14      SBS18      SBS28      SBS37
## 0.02702703 0.04054054 0.02702703 0.68918919 0.13513514 0.52702703 0.04054054
##      SBS40      SBS44
## 0.09459459 0.05405405

```

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

```

##      SBS1      SBS10a     SBS10b      SBS14      SBS18      SBS28      SBS37
## 0.02342633 0.39302667 0.13415977 0.01502674 0.01674129 0.22524153 0.09998130
##      SBS40      SBS44
## 0.03731777 0.05507859

```

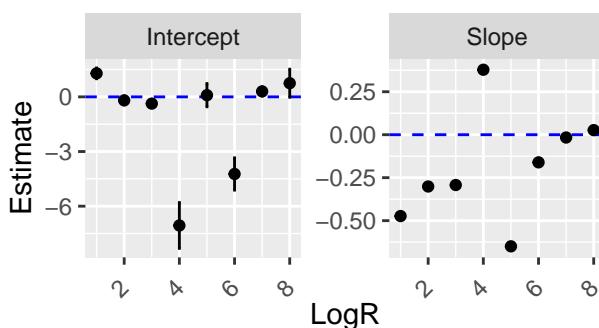
Betas

```
ct <- "ColoRect-AdenoCA"

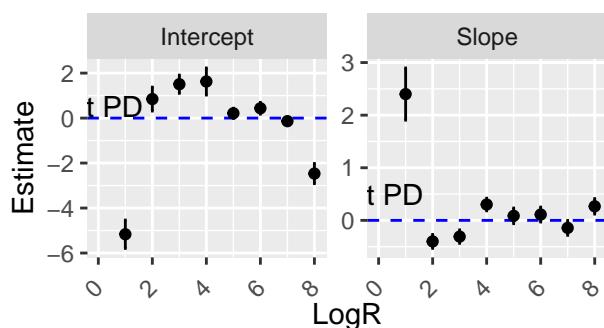
grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)
```

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

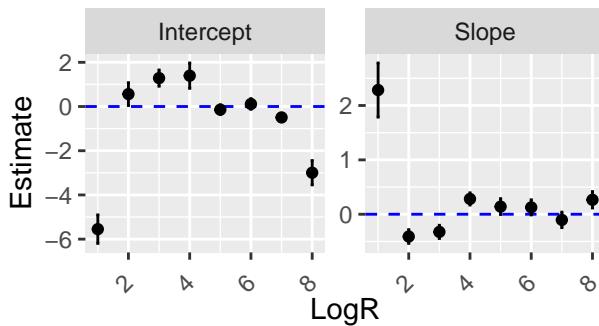
**ColoRect-AdenoCA
diagRE_M**



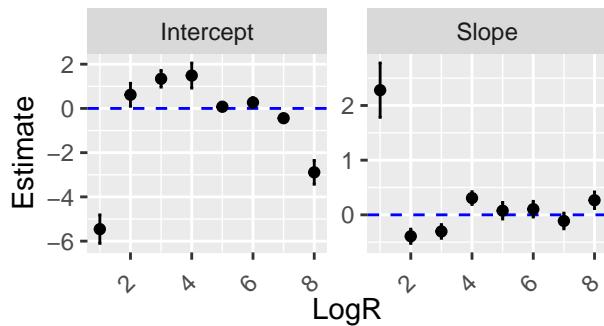
**ColoRect-AdenoCA
fullRE_DMSL**



**ColoRect-AdenoCA
diagRE_DMSL**

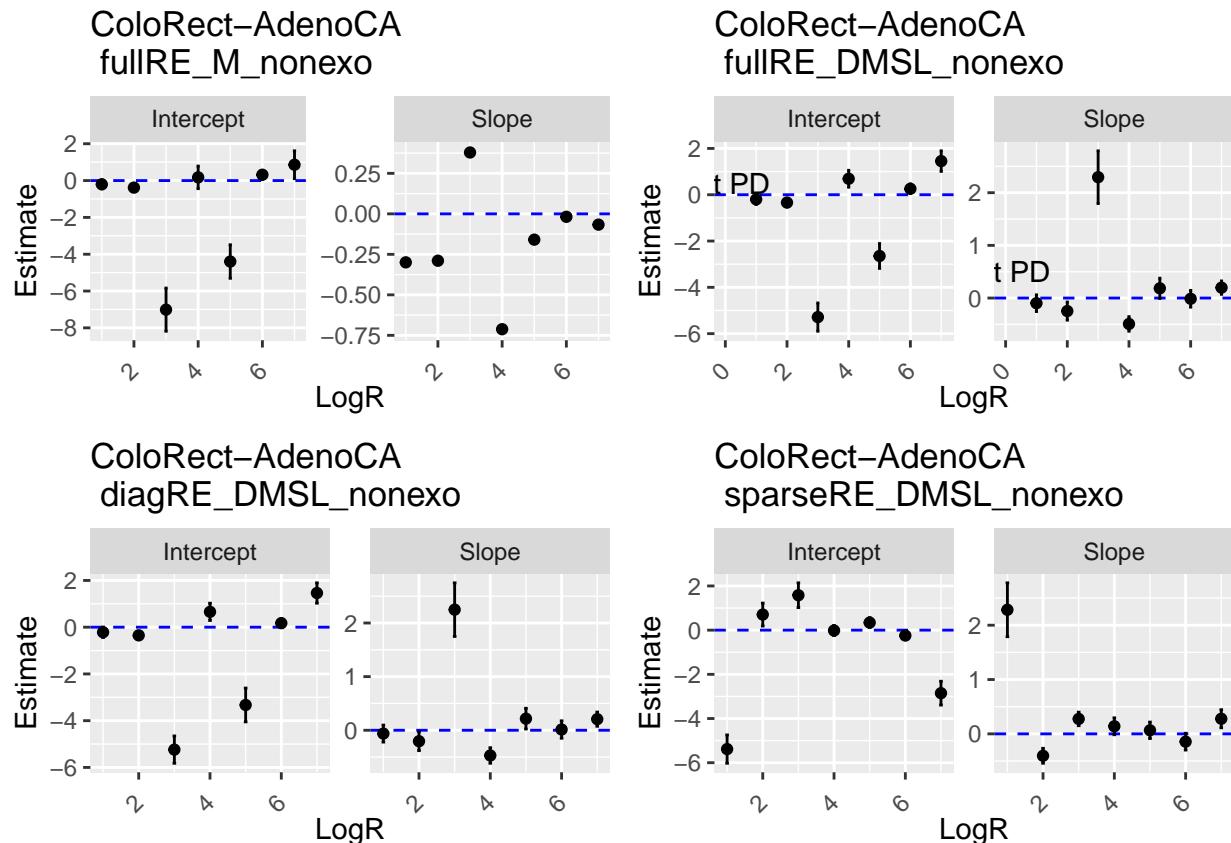


**ColoRect-AdenoCA
sparseRE_DMSL**



```
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

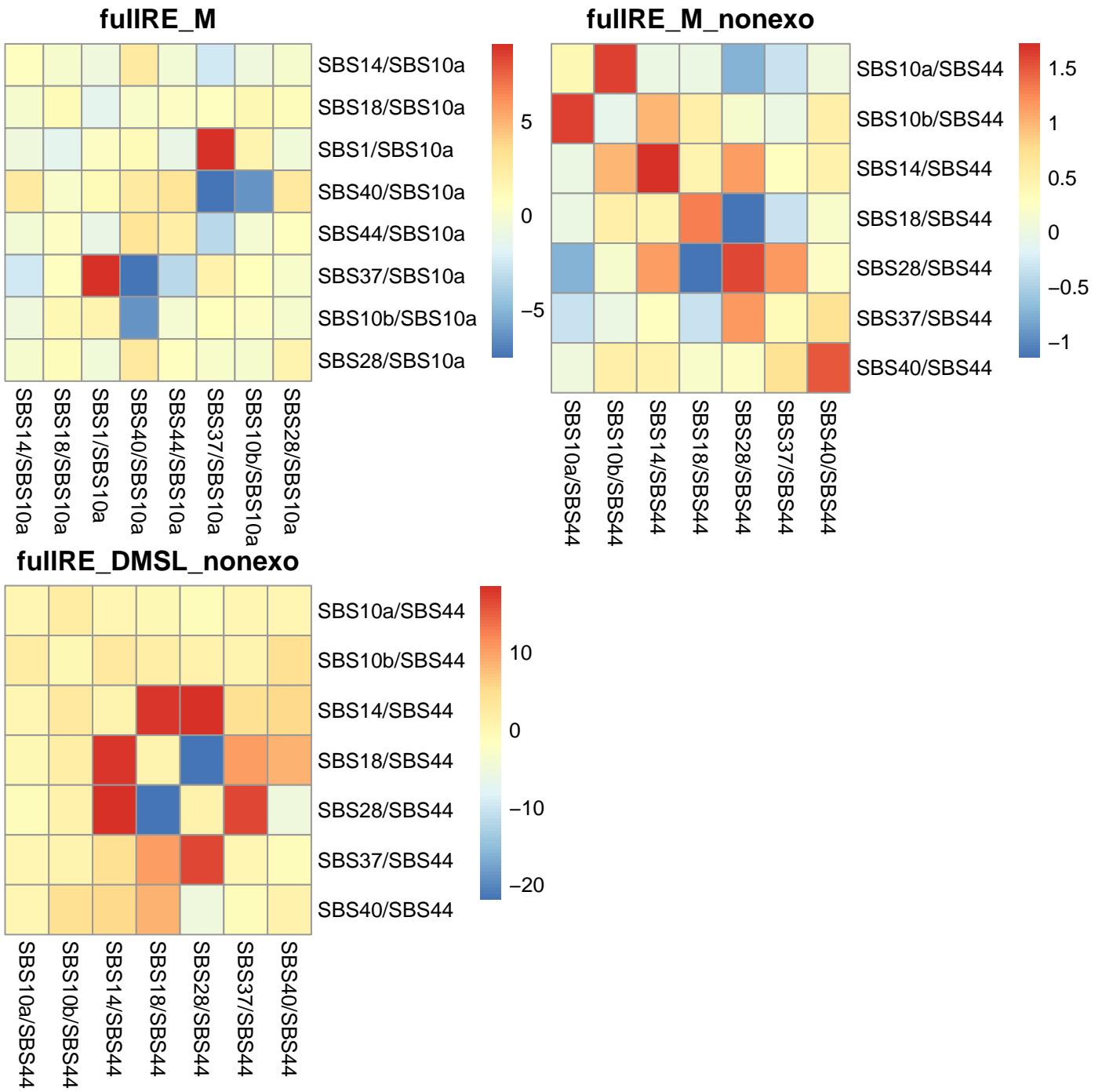
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 RE single lambda DM nonexo to test for differential abundance, giving a p-value of $8.8714208 \times 10^{-16}$.

Covariance matrices



Simulation under inferred data

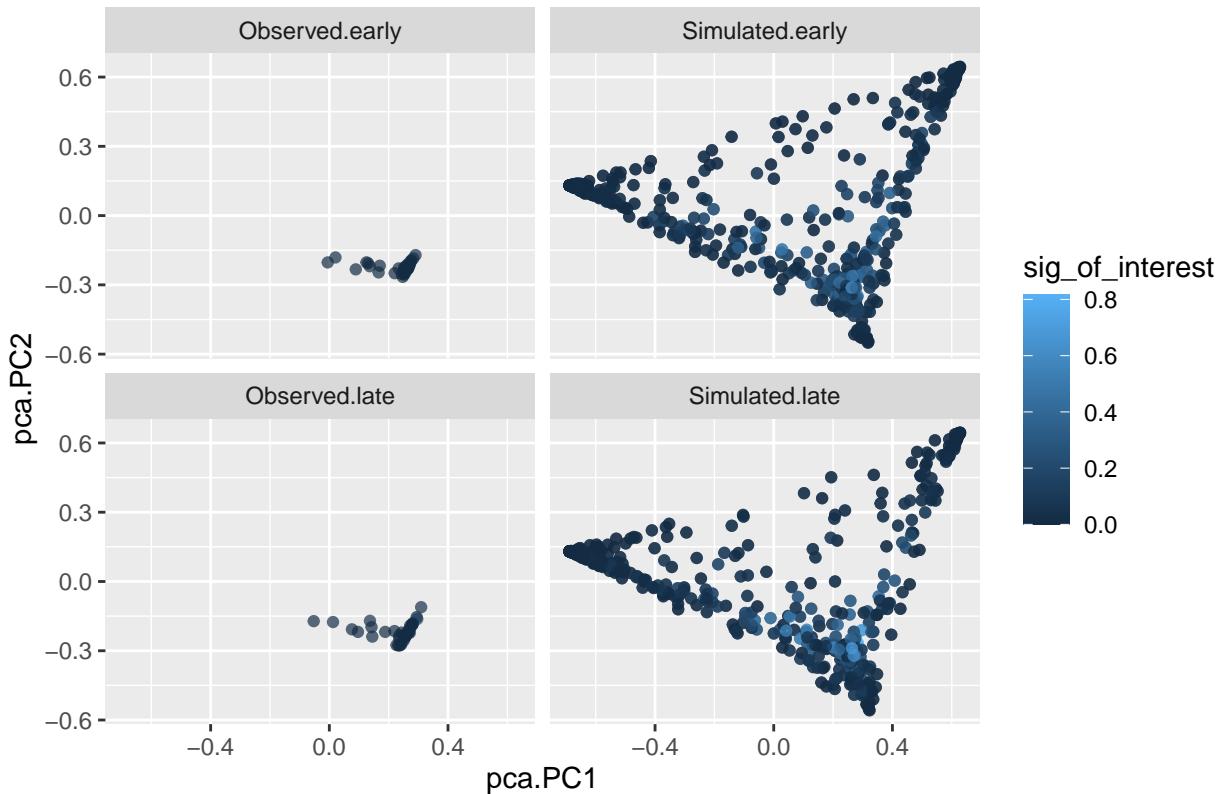
I am simulating using the full effects multinomial, because the function needs to be adapted to diagDMSL.

```
## [1] 37
```

```
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
```

```
## sigma is numerically not positive semidefinite
```

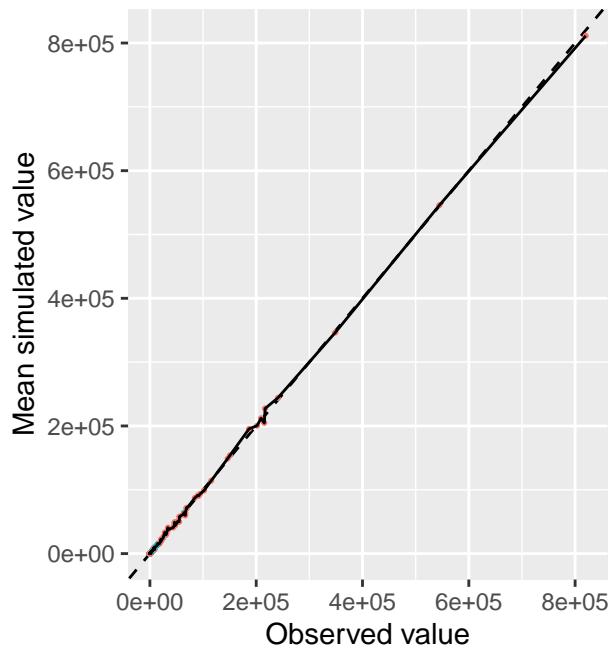
Simulation of ColoRect–AdenoCA samples



Ranked plot for coverage

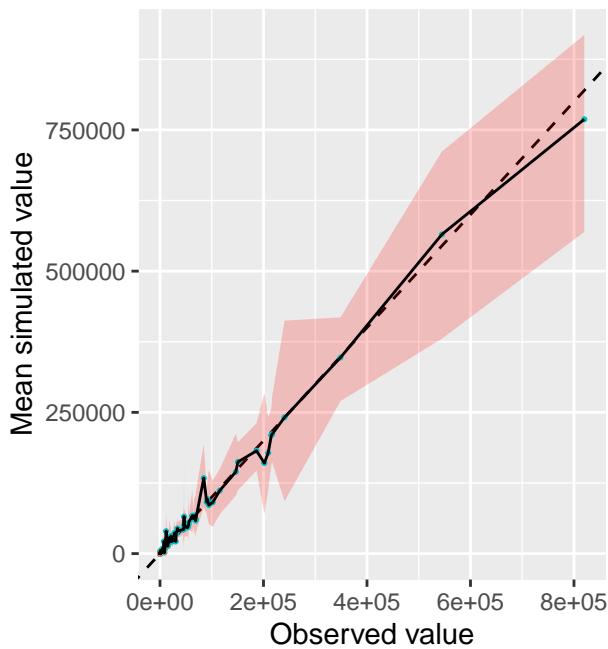
```
ct <- "ColoRect-AdenoCA"
integer_overdispersion_param_DMSL <- 1
obj_ColoRect_AdenoCA_nonexo <- give_subset_sigs_TMBobj(obj_ColoRect_AdenoCA, sigs_to_remove = nonexogenous)
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = full,
data_object = obj_ColoRect_AdenoCA_nonexo,
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_ColoRect_AdenoCA_nonexo,
loglog = F, title = 'obj_ColoRect_AdenoCA (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = diagRE_DMSL_non,
data_object = obj_ColoRect_AdenoCA_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_ColoRect_AdenoCA_nonexo,
loglog = F, title = 'obj_ColoRect_AdenoCA (diag DMSL)'), ncol=2)
```

obj_ColoRect_AdenoCA (M)
FALSE:439; TRUE:153



col ● FALSE ■ TRUE

obj_ColoRect_AdenoCA (diag [
FALSE:147; TRUE:445



col ● FALSE ■ TRUE

Signatures from mutSigExtractor

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

## [1] 37

give_barplot_from_obj(obj = obj_ColoRect_AdenoCA_mutSigExtractor, legend_on = FALSE)

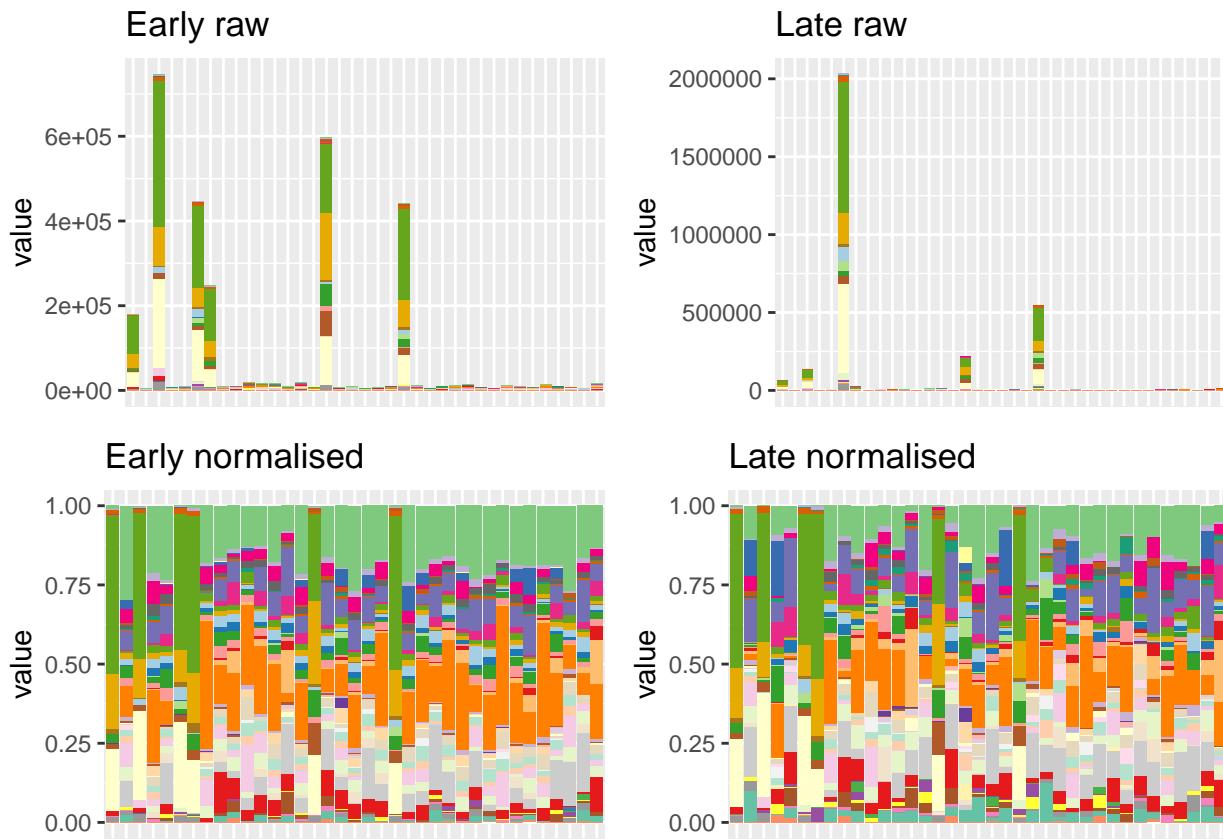
## Creating plot... it might take some time if the data are large. Number of samples: 37
## Creating plot... it might take some time if the data are large. Number of samples: 37
## Creating plot... it might take some time if the data are large. Number of samples: 37
## Creating plot... it might take some time if the data are large. Number of samples: 37

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

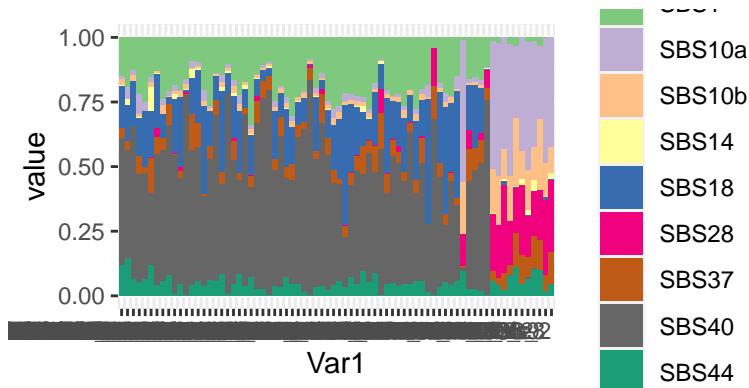
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



Exposures sorted by increasing number of mutations: very clearly there are a few samples with very high number of mutations that also have a completely different mutational signature exposure.

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

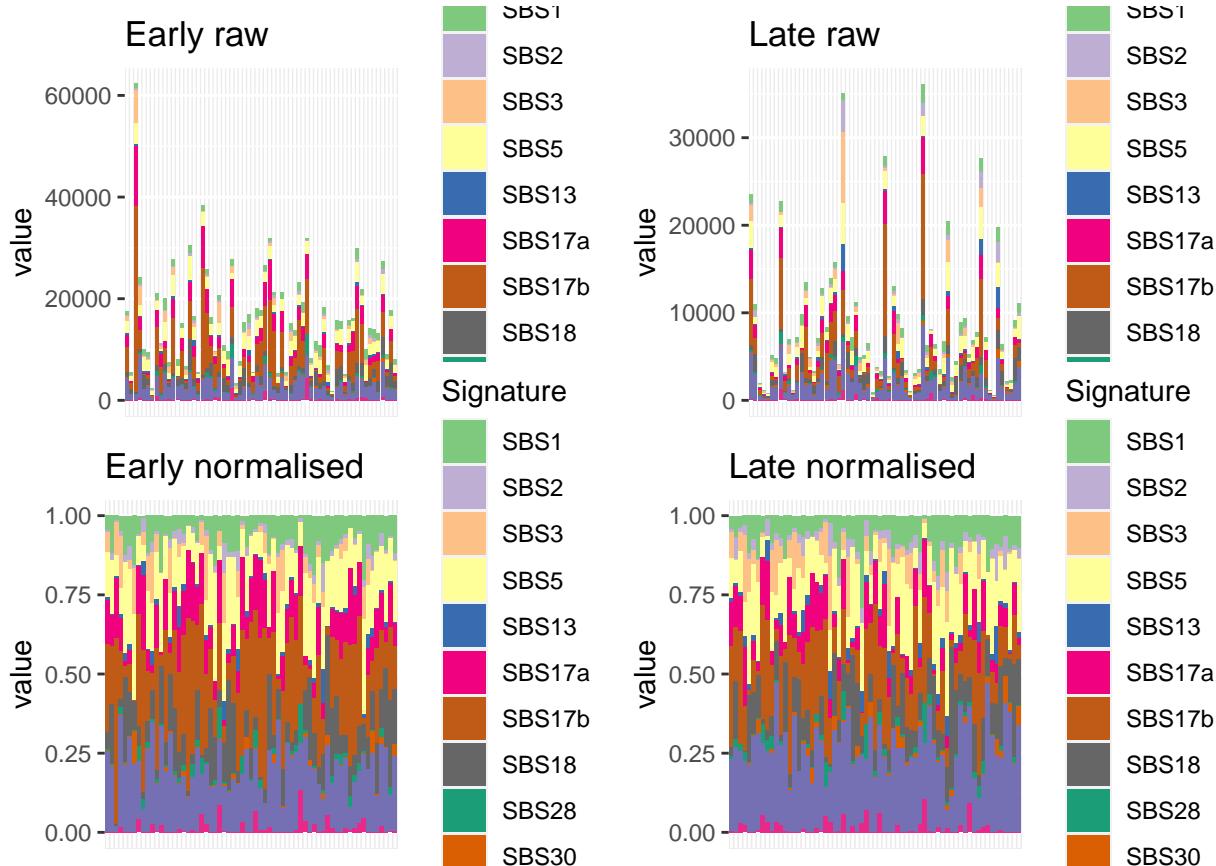
Creating plot... it might take some time if the data are large. Number of samples: 74



Eso-AdenoCA

Barplot and general statistics

```
## [1] 65
## Creating plot... it might take some time if the data are large. Number of samples: 65
## Creating plot... it might take some time if the data are large. Number of samples: 65
## Creating plot... it might take some time if the data are large. Number of samples: 65
## Creating plot... it might take some time if the data are large. Number of samples: 65
```



The number of samples and signatures is:

```
## [1] 130 12
```

The signatures are:

```
## [1] "SBS1"   "SBS2"   "SBS3"   "SBS5"   "SBS13"  "SBS17a" "SBS17b" "SBS18"
## [9] "SBS28"  "SBS30"  "SBS40"  "SBS46"
```

Convergence table

None of the fullRE have converged when including all signatures. When including nonexo, all but fullRE_DMSL_nonexo (using either the highest absolute signature or SBS1) have converged.

```
##           value                      L2                      L1
## 1 Eso-AdenoCA hessian_positivedefinite_bool
## 2 Eso-AdenoCA hessian_nonpositivedefinite_bool
##                                         diagRE_M
##                                         fullRE_M
```

```

## 3 Eso-AdenoCA hessian_nonpositivedefinite_bool diagRE_DMDL
## 4 Eso-AdenoCA hessian_nonpositivedefinite_bool fullRE_halfDM
## 5 Eso-AdenoCA hessian_nonpositivedefinite_bool fullRE_DMDL
## 6 Eso-AdenoCA hessian_positivedefinite_bool diagRE_DMSL
## 7 Eso-AdenoCA hessian_positivedefinite_bool sparseRE_DMSL
## 8 Eso-AdenoCA hessian_nonpositivedefinite_bool fullRE_DMSL
## 9 Eso-AdenoCA hessian_nonpositivedefinite_bool fullRE_DMSL_SBS1
## 10 Eso-AdenoCA hessian_positivedefinite_bool fullRE_M_nonexo
## 11 Eso-AdenoCA hessian_positivedefinite_bool diagRE_DMSL_nonexo
## 12 Eso-AdenoCA hessian_positivedefinite_bool sparseRE_DMSL_nonexo
## 13 Eso-AdenoCA hessian_nonpositivedefinite_bool fullRE_DMSL_nonexo
## 14 Eso-AdenoCA hessian_positivedefinite_bool fullRE_DMDL_nonexo
## 15 Eso-AdenoCA hessian_positivedefinite_bool fullRE_DMDL_sortednonexo

```

Re-running of fitting

Using fullRE_M_nonexo to fit fullRE_DMSL_nonexo

which has a positive-semidefinite covariance matrix, i.e. has converged

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

The fullRE DMSL hasn't, though:

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

Potentially problematic signatures

We notice that there are no truly problematic signatures (SBS30 has the most zeros; 54.6%).

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

```
##      SBS1      SBS2      SBS3      SBS5      SBS13     SBS17a
## 0.000000000 0.023076923 0.392307692 0.000000000 0.215384615 0.038461538
##      SBS17b     SBS18     SBS28     SBS30     SBS40     SBS46
## 0.007692308 0.038461538 0.238461538 0.546153846 0.000000000 0.476923077
```

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

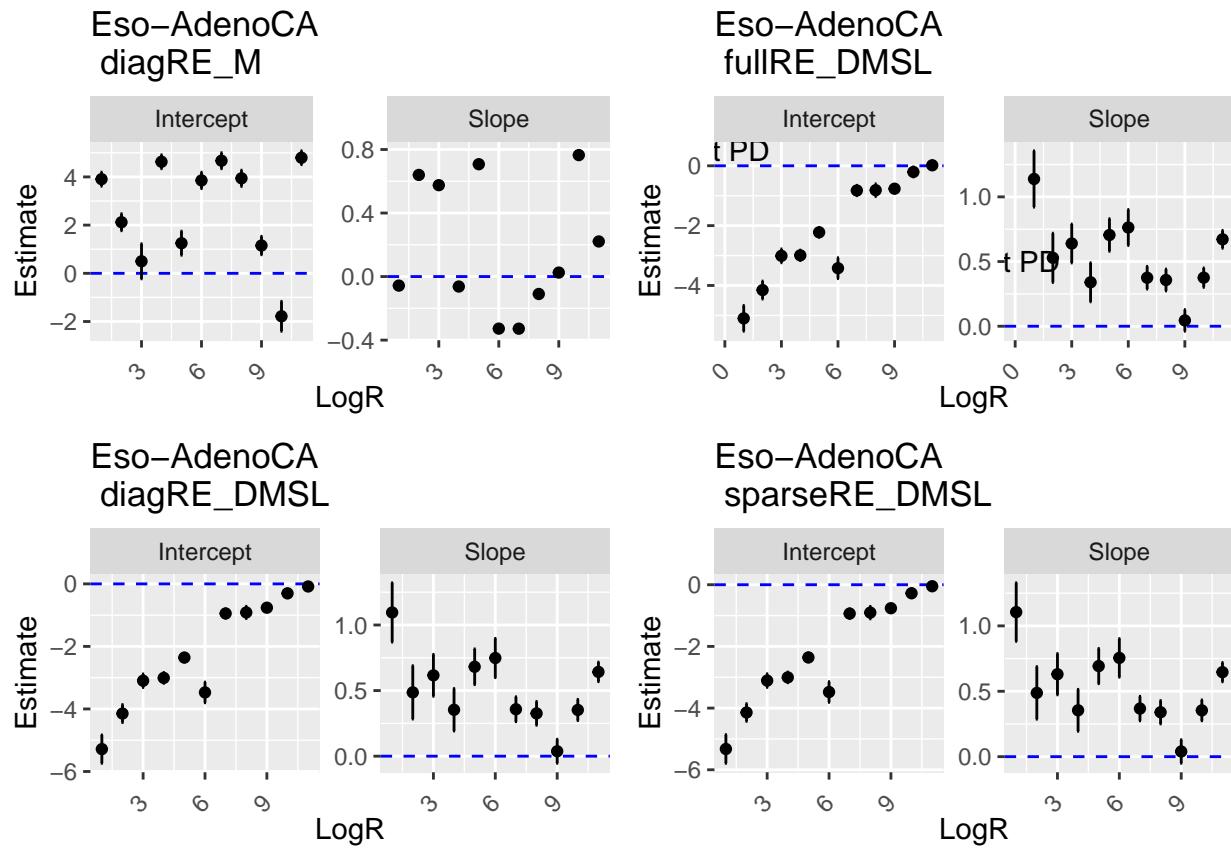
```
##      SBS1      SBS2      SBS3      SBS5      SBS13     SBS17a
## 0.069743929 0.022981455 0.042124010 0.135767687 0.017294837 0.118553858
##      SBS17b     SBS18     SBS28     SBS30     SBS40     SBS46
## 0.265599550 0.088385597 0.020133817 0.006873288 0.198223396 0.014318577
```

Betas

```
ct <- "Eso-AdenoCA"

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
```

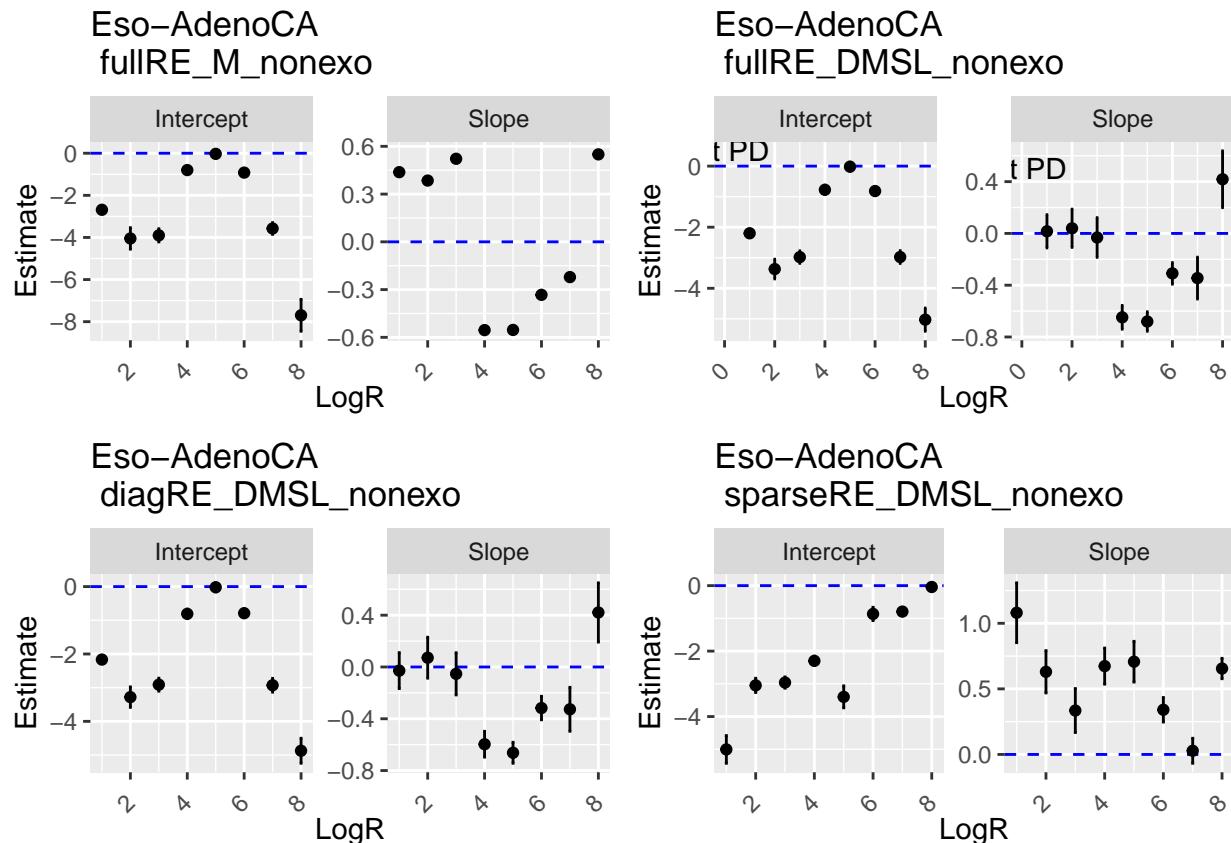


```

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 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

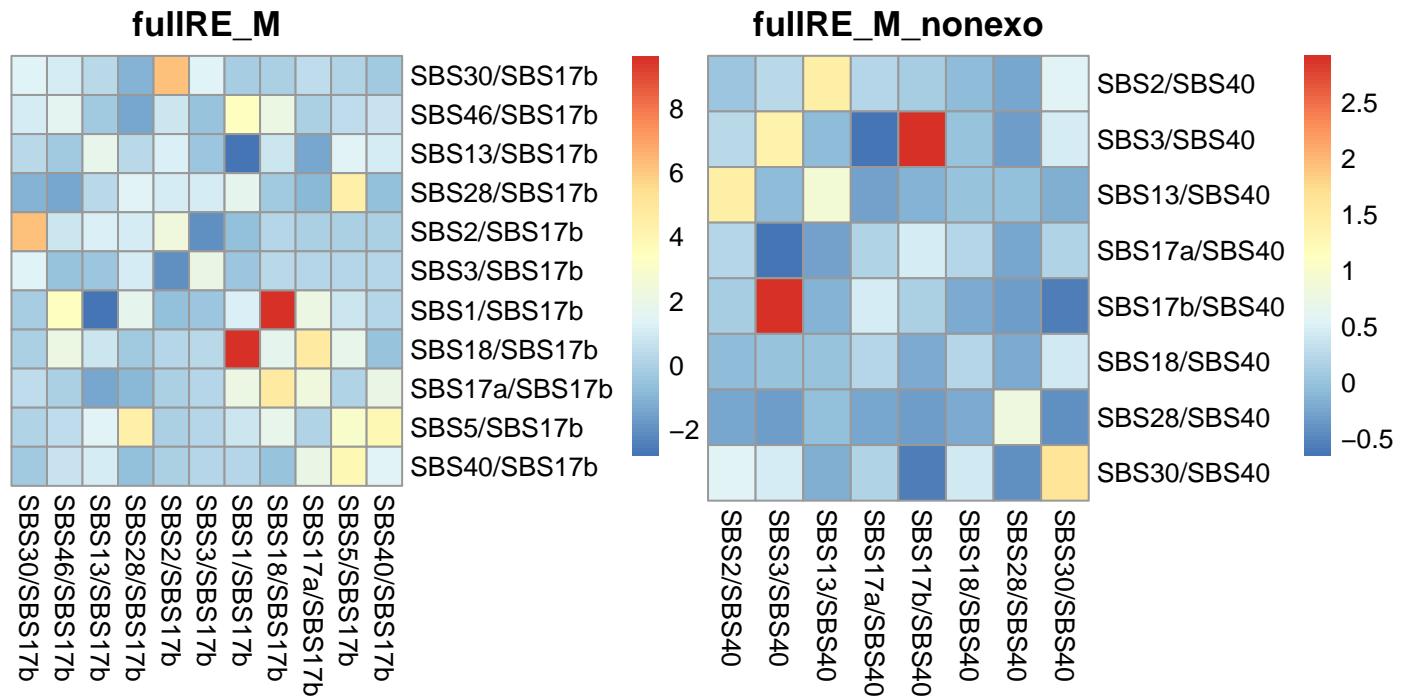
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 diag RE single lambda DM to test for differential abundance, giving a p-value of $2.4465743 \times 10^{-18}$.

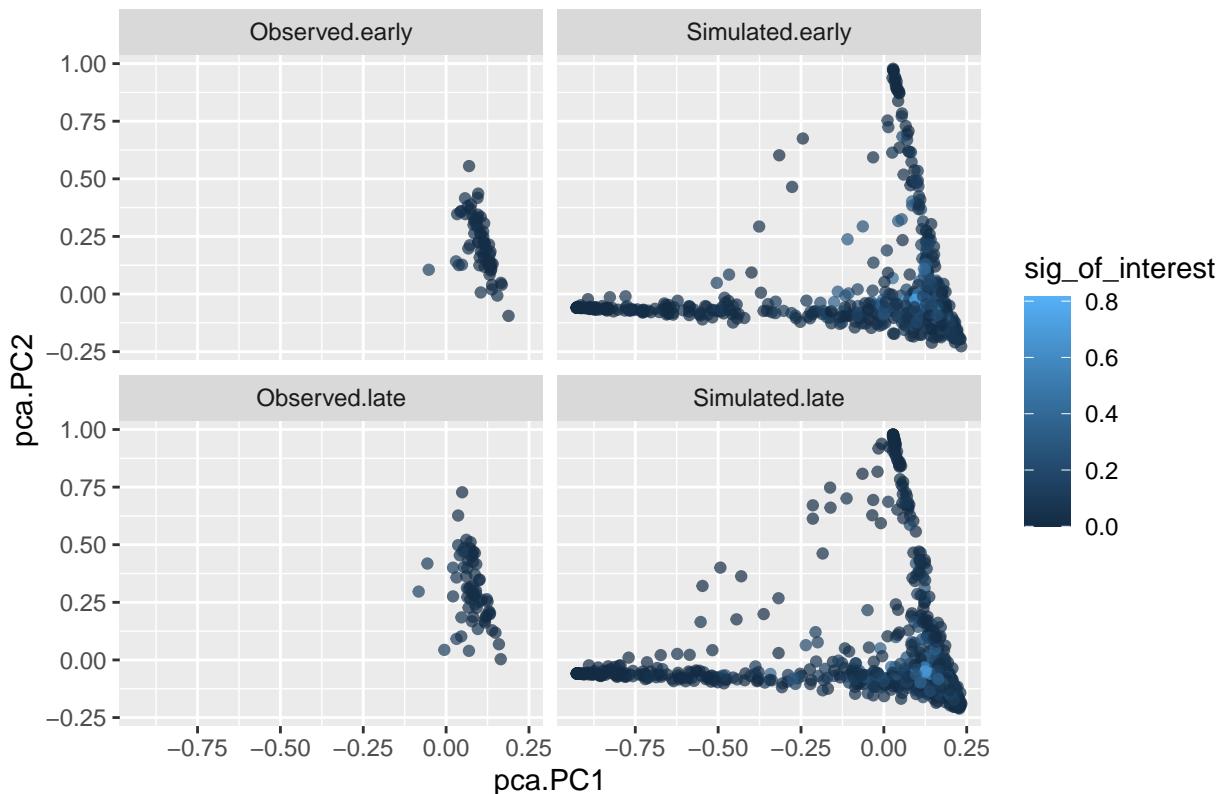
Covariance matrices



Simulation under inferred data

```
## [1] 65
## Warning in .sigma[unlist(sapply(1:(arg_d - 1), function(i) (i - 1) * arg_d + :
## number of items to replace is not a multiple of replacement length
## Warning in .sigma[unlist(sapply(1:(arg_d - 1), function(i) (i) + ((i):(arg_d - :
## number of items to replace is not a multiple of replacement length
## Warning in .sigma[unlist(sapply(1:(arg_d - 1), function(i) (i - 1) * arg_d + :
## number of items to replace is not a multiple of replacement length
## Warning in .sigma[unlist(sapply(1:(arg_d - 1), function(i) (i) + ((i):(arg_d - :
## number of items to replace is not a multiple of replacement length
```

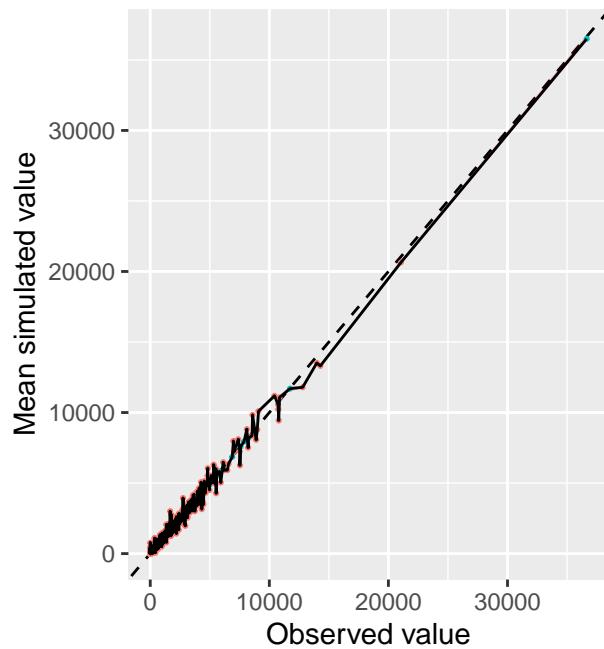
Simulation of Eso–AdenoCA samples



Ranked plot for coverage

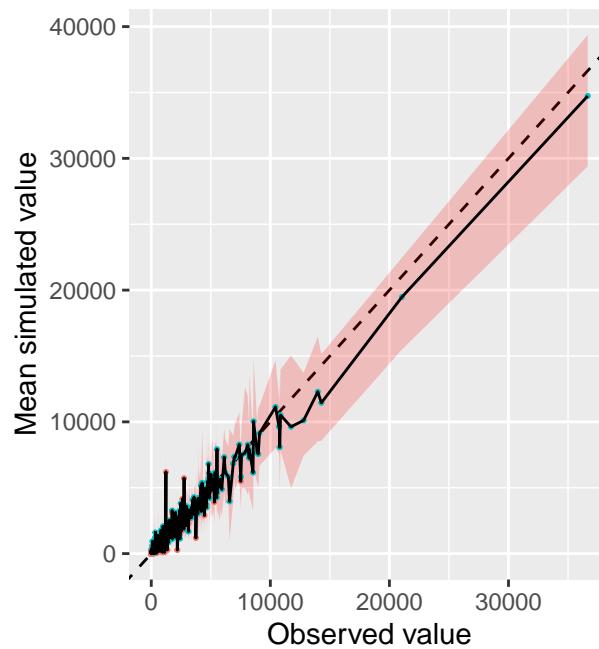
```
ct <- "Eso-AdenoCA"
integer_overdispersion_param_DMSL <- 1
obj_Eso_AdenoCA_nonexo <- give_subset_sigs_TMBobj(obj_Eso_AdenoCA, sigs_to_remove = nonexogenous$V1)
grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = full,
                                         data_object = obj_Eso_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_Eso_AdenoCA_nonexo,
                                         loglog = F, title = 'obj_Eso_AdenoCA (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = diagRE_DMSL_non,
                                         data_object = obj_Eso_AdenoCA_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_Eso_AdenoCA_nonexo,
                                         loglog = F, title = 'obj_Eso_AdenoCA (DMSL)'), ncol=2)
```

obj_Eso_AdenoCA (M)
FALSE:903; TRUE:267



col ● FALSE ● TRUE

obj_Eso_AdenoCA (DMSL)
FALSE:247; TRUE:923



col ● FALSE ● TRUE

Signatures from mutSigExtractor

The signatures from mutSigExtractor are as follows:

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

## [1] 65
give_barplot_from_obj(obj = obj_Eso_AdenoCA_mutSigExtractor, legend_on = FALSE)

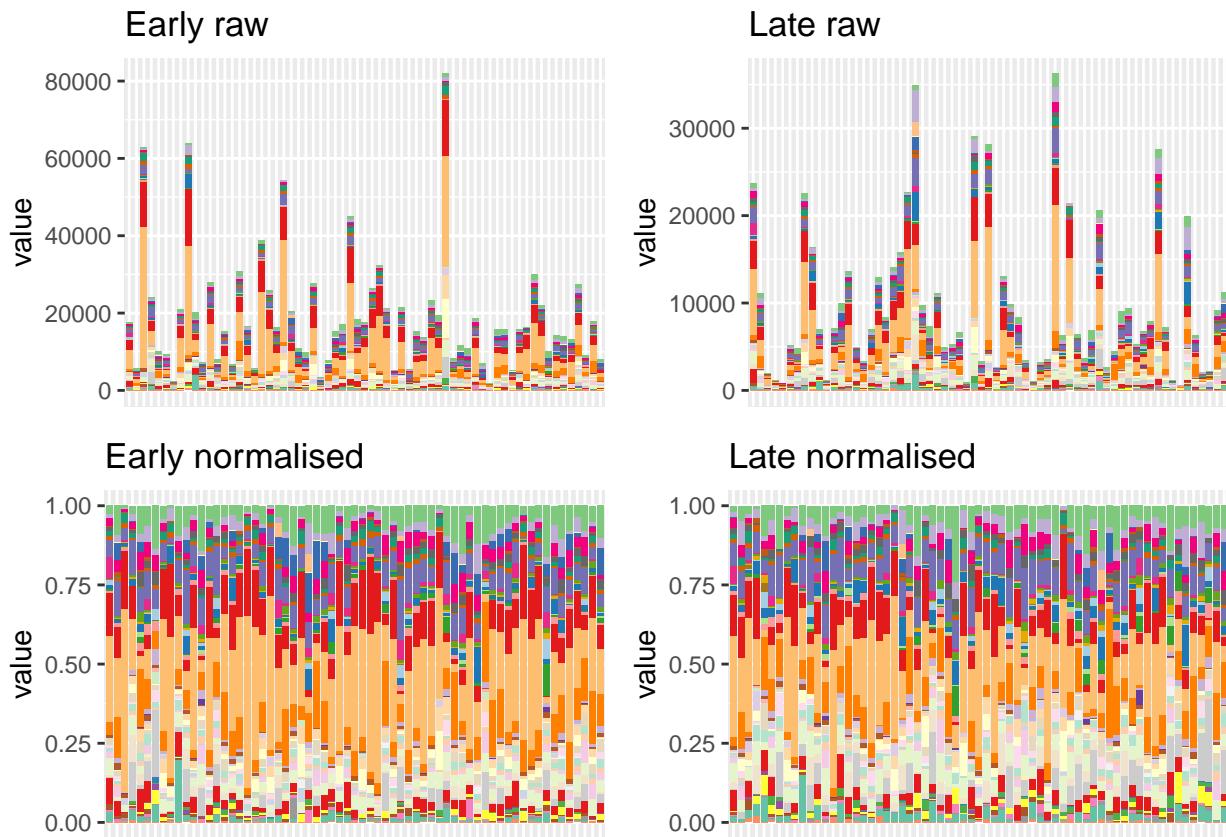
## Creating plot... it might take some time if the data are large. Number of samples: 65
## Creating plot... it might take some time if the data are large. Number of samples: 65
## Creating plot... it might take some time if the data are large. Number of samples: 65
## Creating plot... it might take some time if the data are large. Number of samples: 65

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

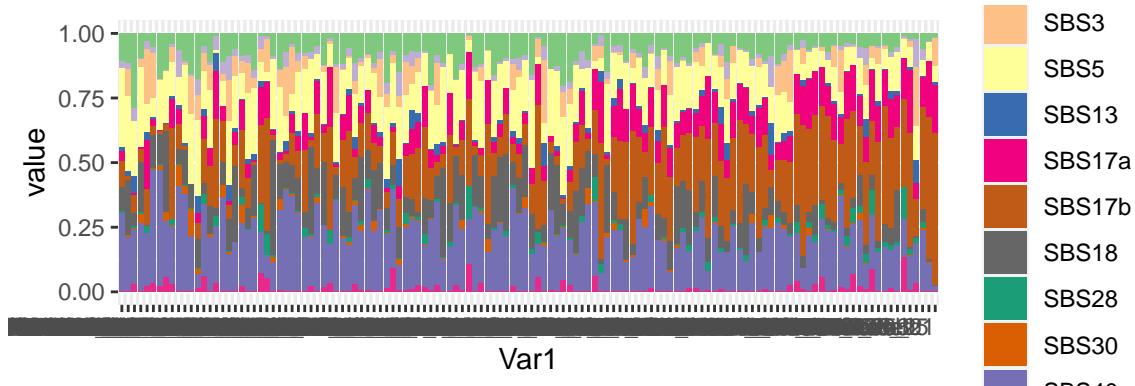
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



Exposures sorted by increasing number of mutations: there is a trend of samples with more mutations having more SBS17b and less SBS5, relatively.

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

Creating plot... it might take some time if the data are large. Number of samples: 130

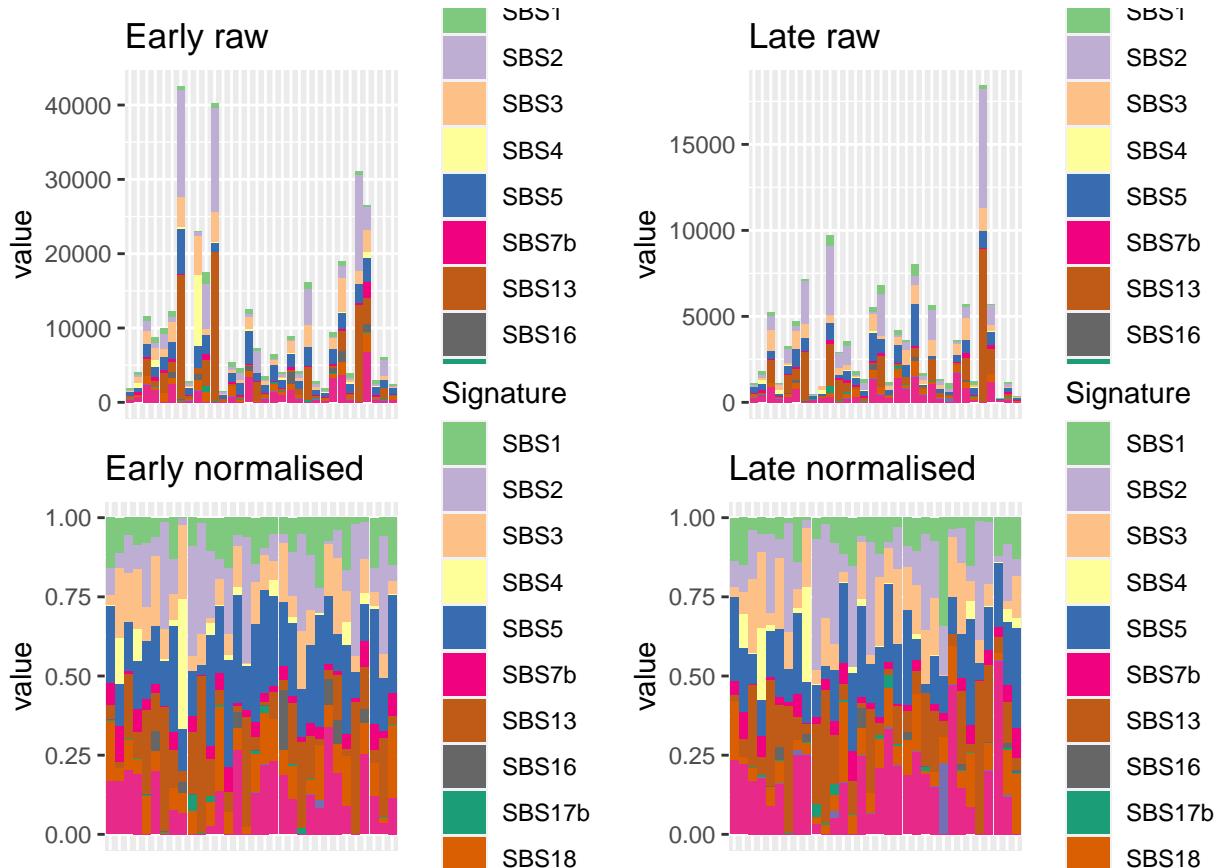


Head-SCC

Barplot and general statistics

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

```
## Creating plot... it might take some time if the data are large. Number of samples: 32
## Creating plot... it might take some time if the data are large. Number of samples: 32
## Creating plot... it might take some time if the data are large. Number of samples: 32
## Creating plot... it might take some time if the data are large. Number of samples: 32
```



The number of samples and signatures is:

```
## [1] 64 12
```

The signatures are:

```
## [1] "SBS1"   "SBS2"   "SBS3"   "SBS4"   "SBS5"   "SBS7b"  "SBS13"  "SBS16"
## [9] "SBS17b" "SBS18"  "SBS33"  "SBS40"
```

Convergence table

We don't have converged results for the multinomial with full RE, but for nonexogenous signatures everything has.

##	value	L2	L1
## 1	Head-SCC hessian_positivedefinite_bool	diagRE_M	
## 2	Head-SCC hessian_nonpositivedefinite_bool	fullRE_M	

```

## 3 Head-SCC hessian_nonpositivedefinite_bool          diagRE_DMDL
## 4 Head-SCC hessian_nonpositivedefinite_bool          fullRE_halfDM
## 5 Head-SCC hessian_nonpositivedefinite_bool          fullRE_DMDL
## 6 Head-SCC    hessian_positivedefinite_bool          diagRE_DMSL
## 7 Head-SCC    hessian_positivedefinite_bool          sparseRE_DMSL
## 8 Head-SCC hessian_nonpositivedefinite_bool          fullRE_DMSL
## 9 Head-SCC hessian_nonpositivedefinite_bool          fullRE_DMSL_SBS1
## 10 Head-SCC   hessian_positivedefinite_bool          fullRE_M_nonexo
## 11 Head-SCC   hessian_positivedefinite_bool          diagRE_DMSL_nonexo
## 12 Head-SCC   hessian_positivedefinite_bool          sparseRE_DMSL_nonexo
## 13 Head-SCC   hessian_positivedefinite_bool          fullRE_DMSL_nonexo
## 14 Head-SCC hessian_nonpositivedefinite_bool          fullRE_DMDL_nonexo
## 15 Head-SCC                                     Timeout fullRE_DMDL_sortednonexo

```

Re-running of fitting

We don't need refitting, as the results have already converged.

Potentially problematic signatures

SBS33 is likely to be problematic.

```

colSums(obj_Head(SCC$Y == 0)/nrow(obj_Head(SCC$Y)

##      SBS1     SBS2     SBS3     SBS4     SBS5     SBS7b     SBS13     SBS16     SBS17b     SBS18
## 0.00000 0.00000 0.06250 0.50000 0.00000 0.06250 0.00000 0.75000 0.40625 0.09375
##      SBS33     SBS40
## 0.81250 0.21875

colSums(obj_Head(SCC$Y)/sum(obj_Head(SCC$Y)

##      SBS1     SBS2     SBS3     SBS4     SBS5     SBS7b
## 0.052157398 0.209133263 0.121874082 0.030243442 0.152377754 0.025011542
##      SBS13     SBS16     SBS17b     SBS18     SBS33     SBS40
## 0.225057712 0.017861490 0.005867786 0.056873033 0.001013641 0.102528856

```

Betas

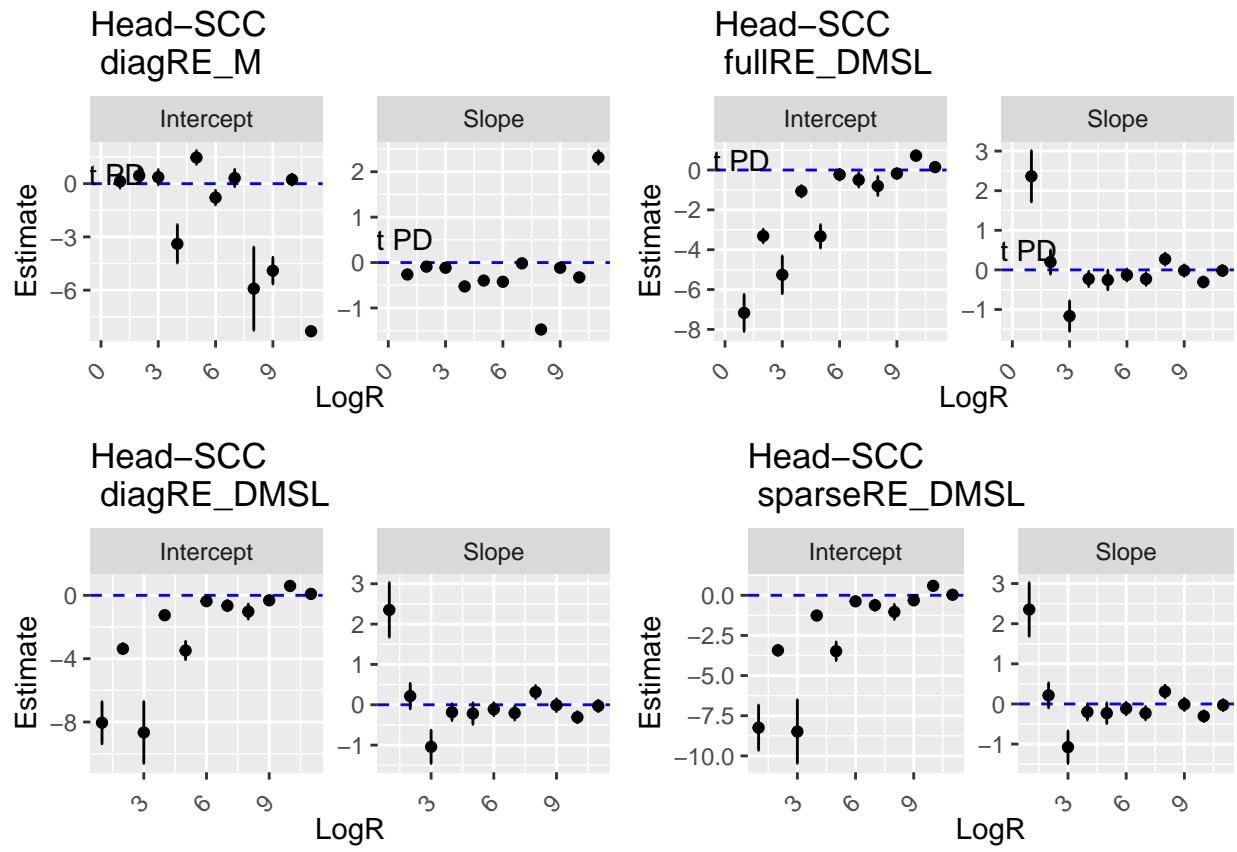
```

ct <- "Head-SCC"

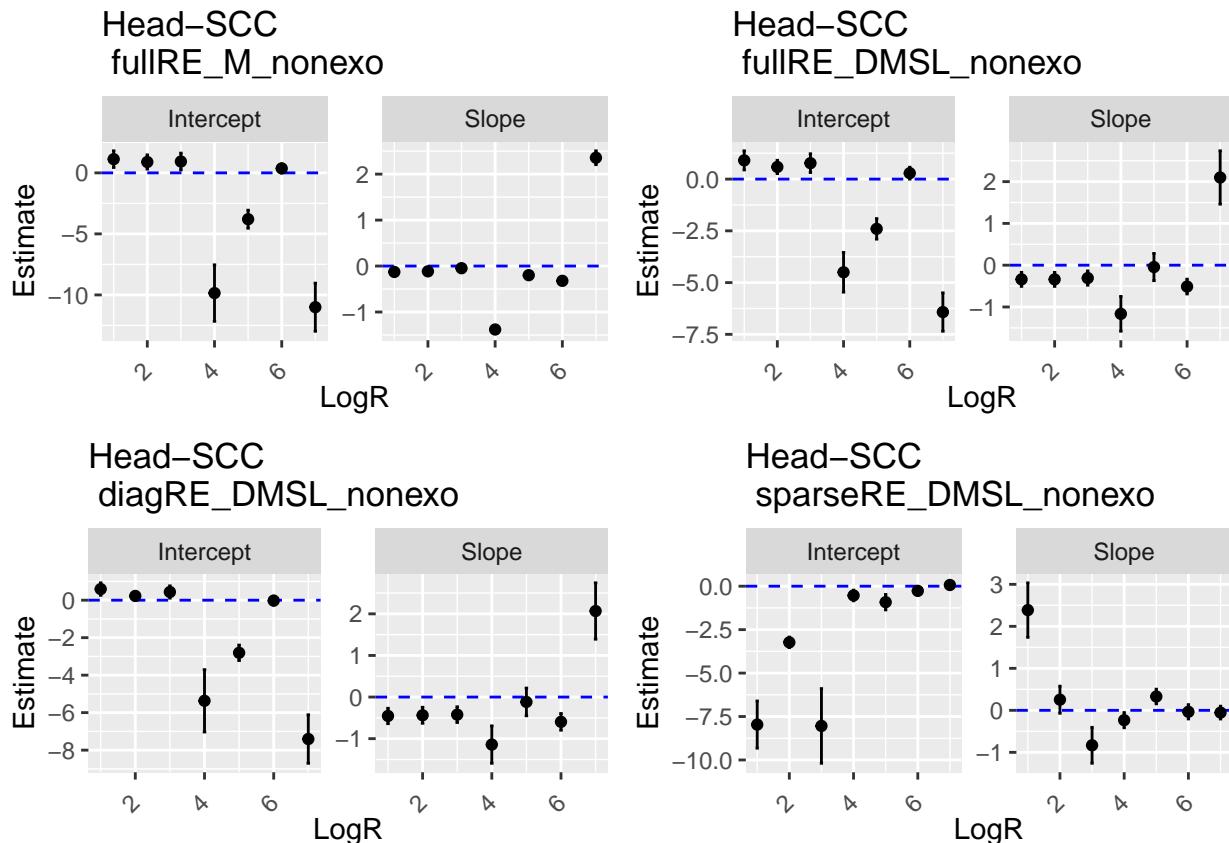
grid.arrange(plot_betas(fullRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
             plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
             plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
             plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct,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

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

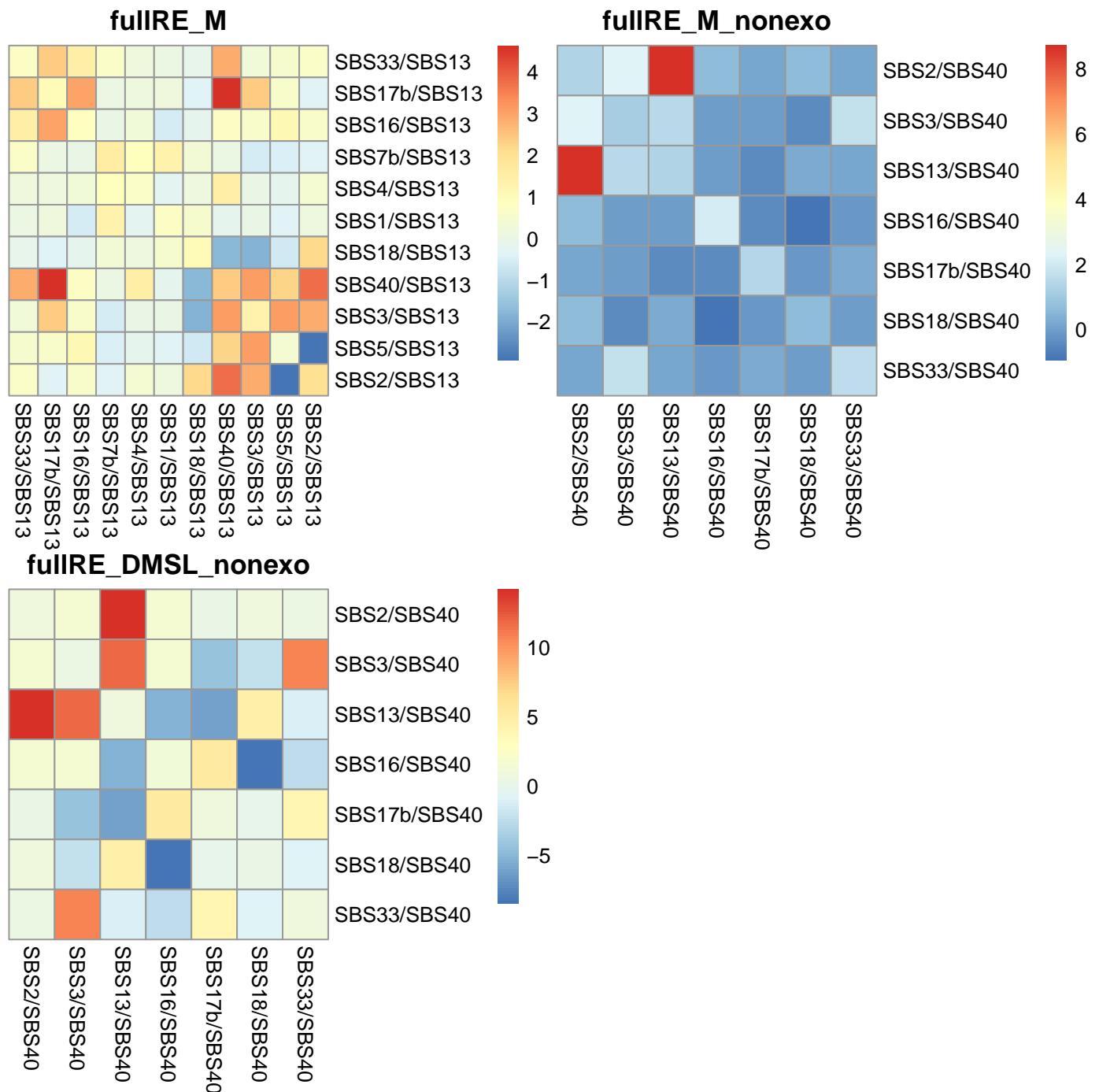
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 8.4420109×10^{-5} .

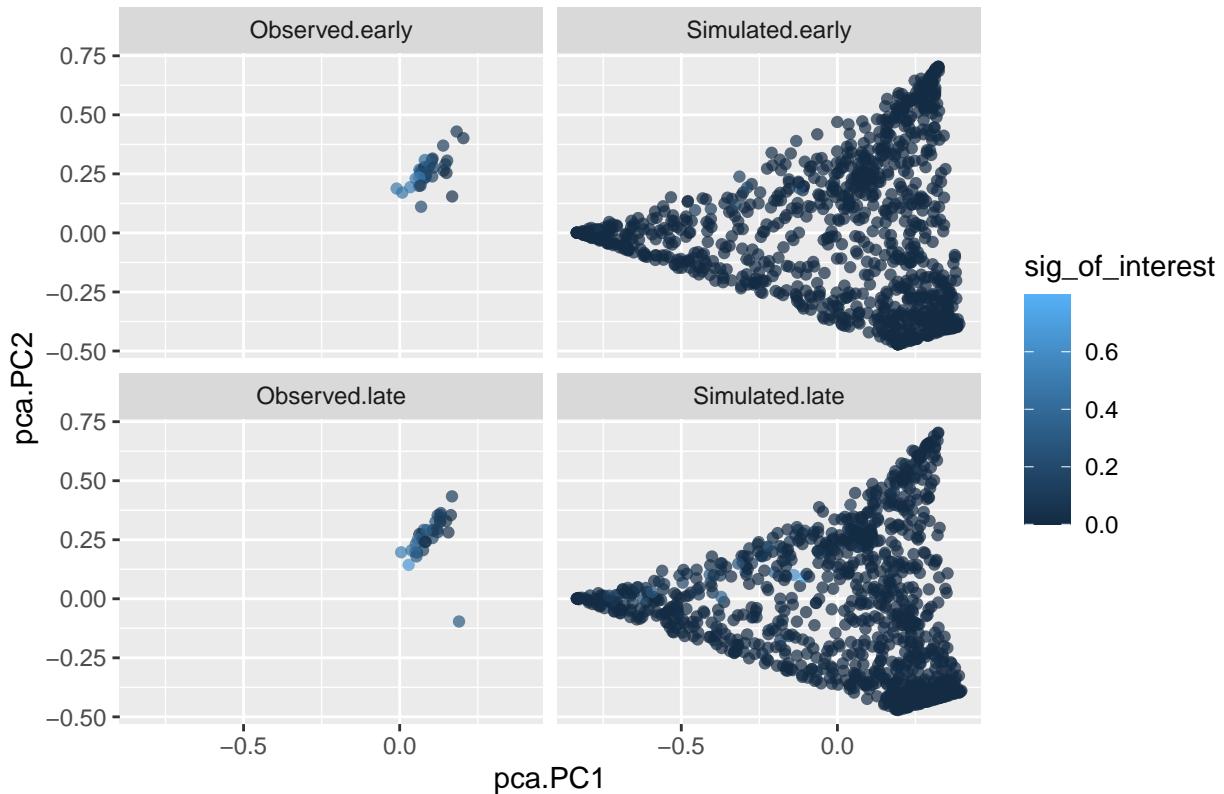
Covariance matrices



Simulation under inferred data

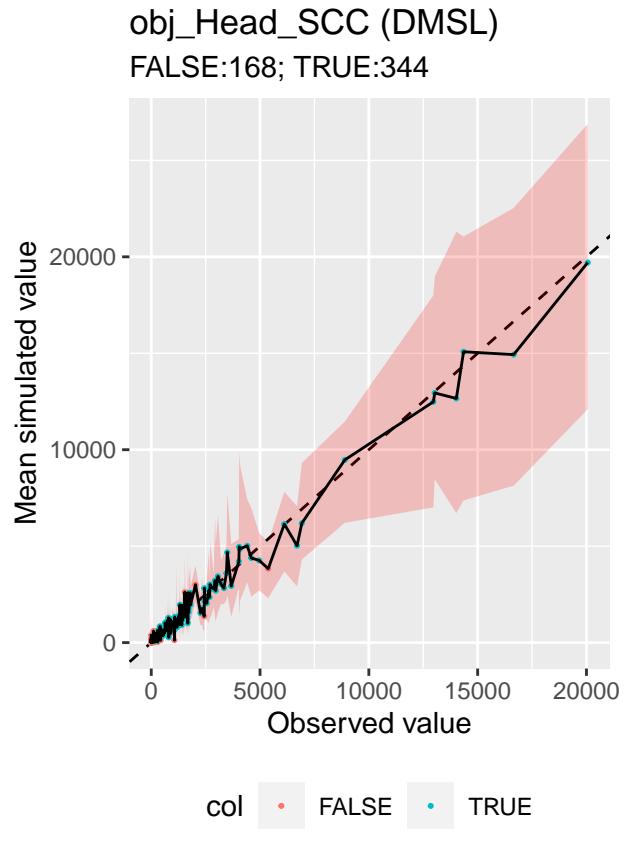
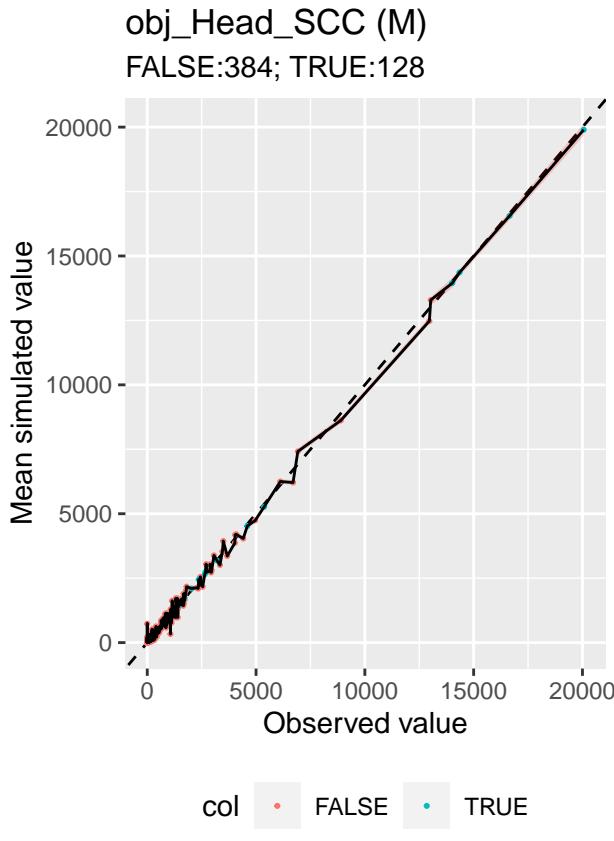
```
## [1] 32
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

Simulation of Head–SCC samples



Ranked plot for coverage

```
ct <- "Head-SCC"
integer_overdispersion_param_DMSL <- 1
obj_Head_SCC_nonexo <- give_subset_sigs_TMBobj(obj_Head_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_Head_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_Head_SCC_nonexo,
loglog = F, title = 'obj_Head_SCC (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_Head_SCC_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_Head_SCC_nonexo,
loglog = F, title = 'obj_Head_SCC (DMSL)'), ncol=2)
```



Signatures from mutSigExtractor

The signatures from mutSigExtractor are as follows:

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

## [1] 32

give_barplot_from_obj(obj = obj_Head_SCC_mutSigExtractor, legend_on = FALSE)

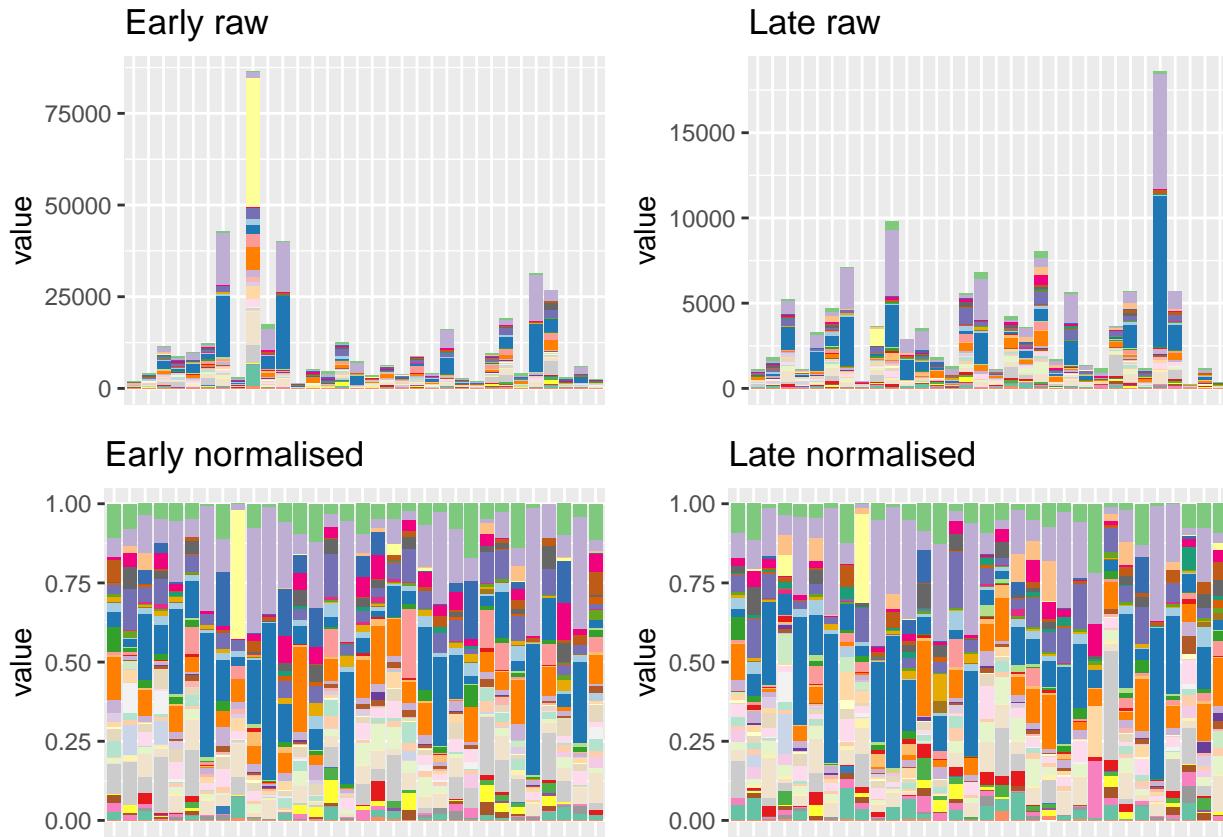
## Creating plot... it might take some time if the data are large. Number of samples: 32
## Creating plot... it might take some time if the data are large. Number of samples: 32
## Creating plot... it might take some time if the data are large. Number of samples: 32
## Creating plot... it might take some time if the data are large. Number of samples: 32

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

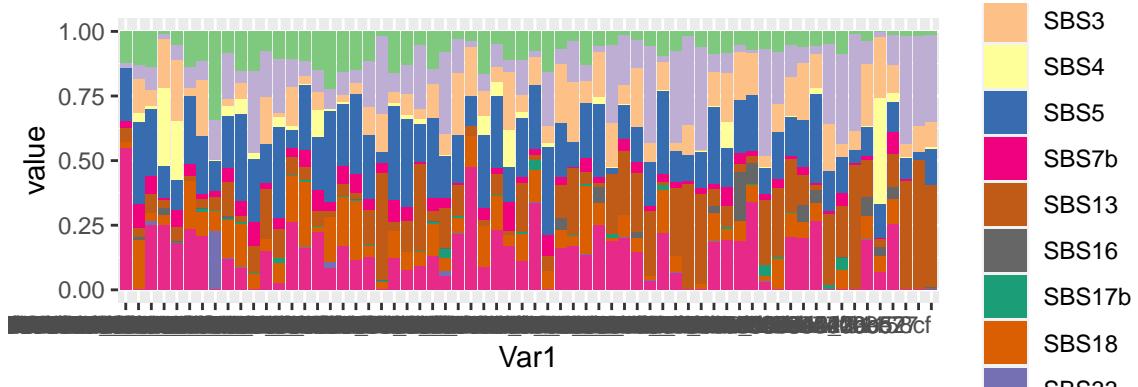
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



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_Head_SCC$Y)),
              order_labels = names(sort(rowSums(non_duplicated_rows(obj_Head_SCC$Y)),
                                         decreasing = F)))
```

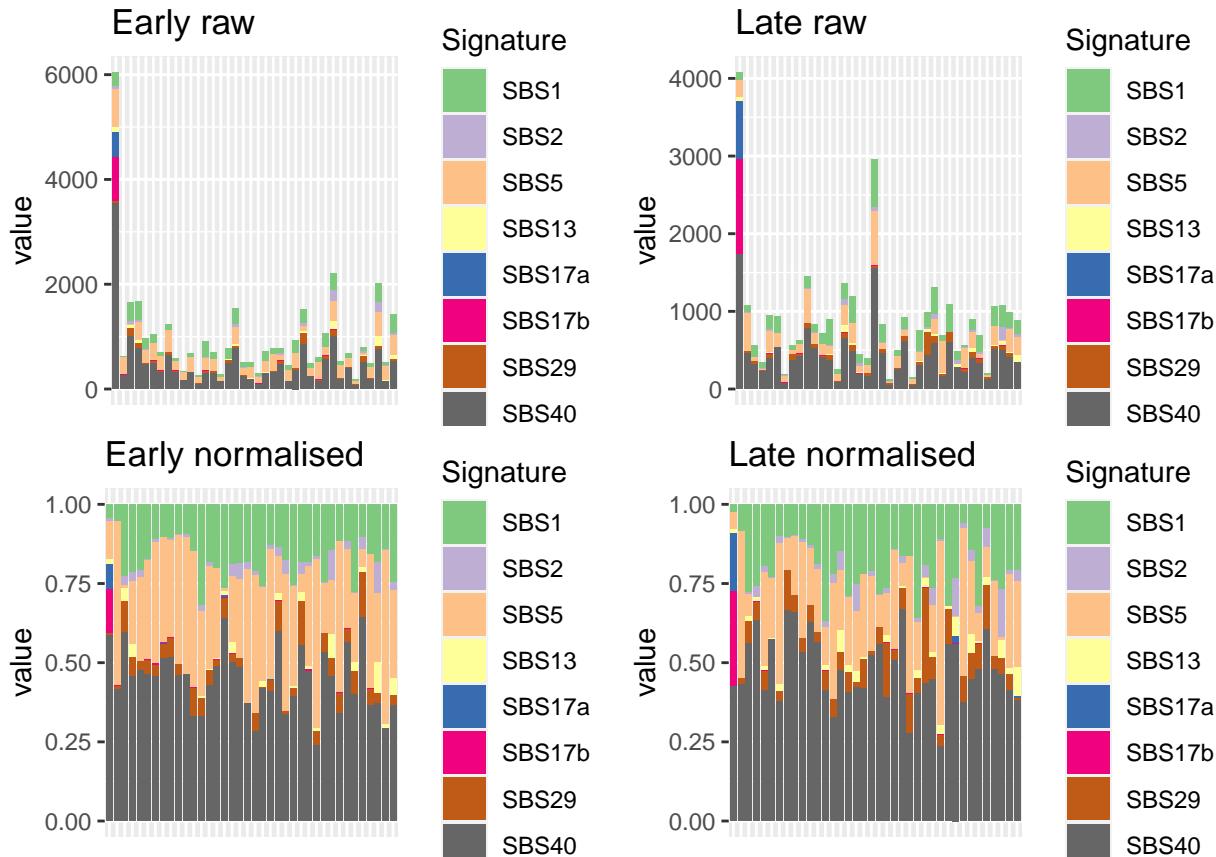
Creating plot... it might take some time if the data are large. Number of samples: 64



Kidney-ChRCC

Barplot and general statistics

```
## [1] 38
## Creating plot... it might take some time if the data are large. Number of samples: 38
## Creating plot... it might take some time if the data are large. Number of samples: 38
## Creating plot... it might take some time if the data are large. Number of samples: 38
## Creating plot... it might take some time if the data are large. Number of samples: 38
```



The number of samples and signatures is:

```
## [1] 76 8
```

The signatures are:

```
## [1] "SBS1"   "SBS2"   "SBS5"   "SBS13"  "SBS17a" "SBS17b" "SBS29"  "SBS40"
```

Convergence table

For all signatures, no fullRE model has converged. For nonexogenous ones, all have.

	L2	L1
## 1 Kidney-ChRCC hessian_nonpositivedefinite_bool		diagRE_M
## 2 Kidney-ChRCC hessian_nonpositivedefinite_bool		fullRE_M
## 3 Kidney-ChRCC hessian_nonpositivedefinite_bool		diagRE_DMDL
## 4 Kidney-ChRCC hessian_nonpositivedefinite_bool		fullRE_halfDM

```

## 5 Kidney-ChRCC hessian_nonpositivedefinite_bool fullRE_DMDL
## 6 Kidney-ChRCC hessian_positivedefinite_bool diagRE_DMSL
## 7 Kidney-ChRCC hessian_positivedefinite_bool sparseRE_DMSL
## 8 Kidney-ChRCC hessian_nonpositivedefinite_bool fullRE_DMSL
## 9 Kidney-ChRCC hessian_nonpositivedefinite_bool fullRE_DMSL_SBS1
## 10 Kidney-ChRCC hessian_positivedefinite_bool fullRE_M_nonexo
## 11 Kidney-ChRCC hessian_positivedefinite_bool diagRE_DMSL_nonexo
## 12 Kidney-ChRCC hessian_positivedefinite_bool sparseRE_DMSL_nonexo
## 13 Kidney-ChRCC hessian_positivedefinite_bool fullRE_DMSL_nonexo
## 14 Kidney-ChRCC hessian_nonpositivedefinite_bool fullRE_DMDL_nonexo
## 15 Kidney-ChRCC hessian_nonpositivedefinite_bool fullRE_DMDL_sortednonexo

```

Re-running of fitting

We do not need to re-run any model fitting.

Potentially problematic signatures

We notice that SBS17a and SBS17b are perhaps problematic.

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

```

##      SBS1      SBS2      SBS5      SBS13     SBS17a     SBS17b     SBS29
## 0.00000000 0.23684211 0.05263158 0.35526316 0.89473684 0.89473684 0.09210526
##      SBS40
## 0.00000000

```

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

```

##      SBS1      SBS2      SBS5      SBS13     SBS17a     SBS17b     SBS29
## 0.17183661 0.02350905 0.21046460 0.02066116 0.01747822 0.02920482 0.04789759
##      SBS40
## 0.47894796

```

Betas

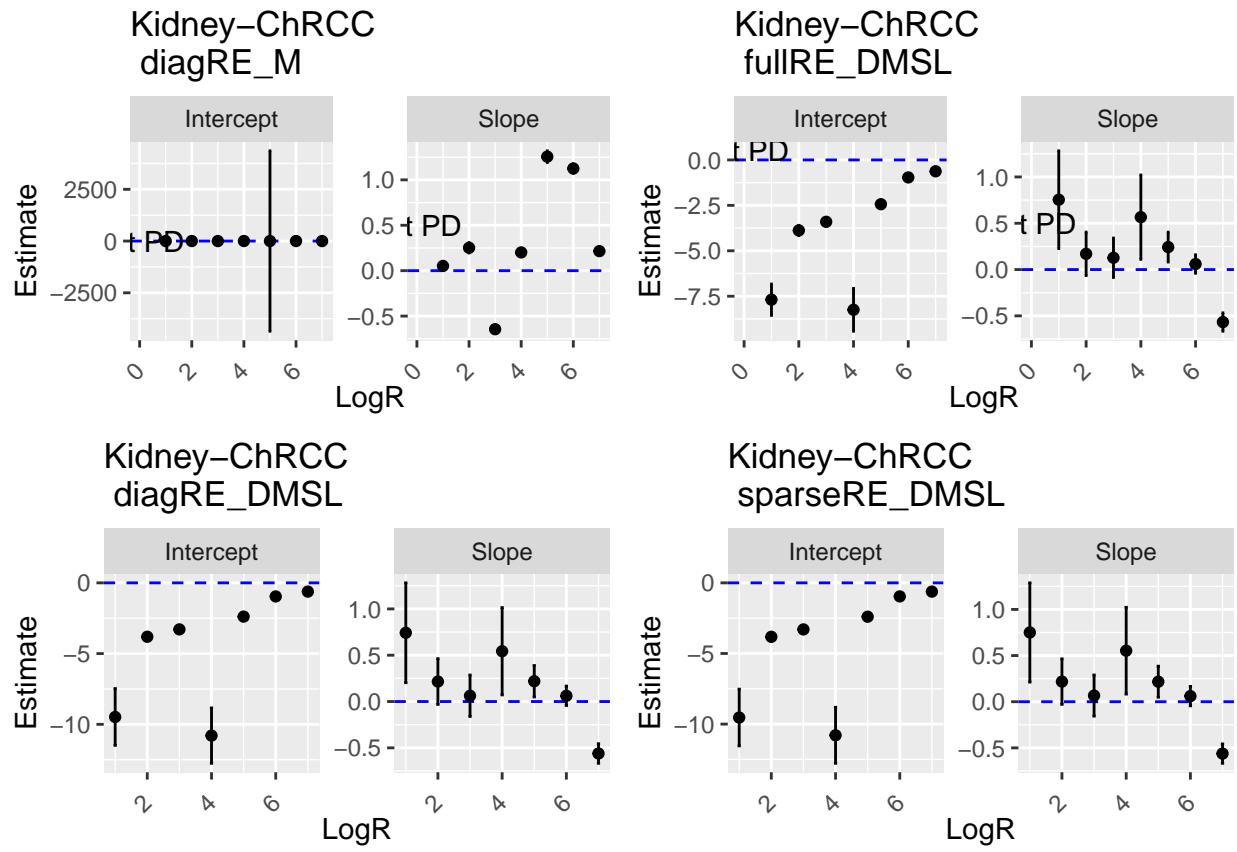
```

ct <- "Kidney-ChRCC"

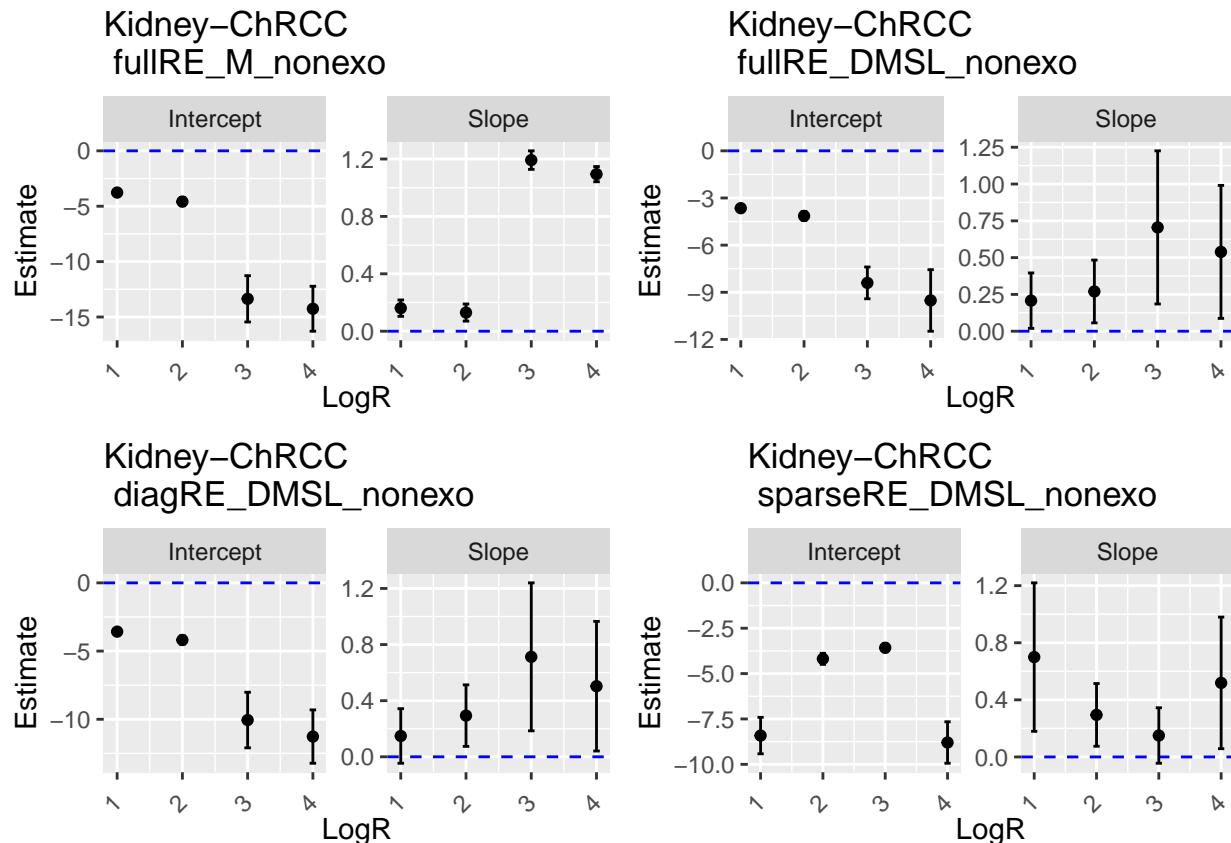
grid.arrange(plot_betas(diagRE_M[[ct]])+ggtitle(paste0(ct, '\n diagRE_M')),
plot_betas(fullRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n fullRE_DMSL')),
plot_betas(diagRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL')),
plot_betas(sparseRE_DMSL[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL')), nrow=2)

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

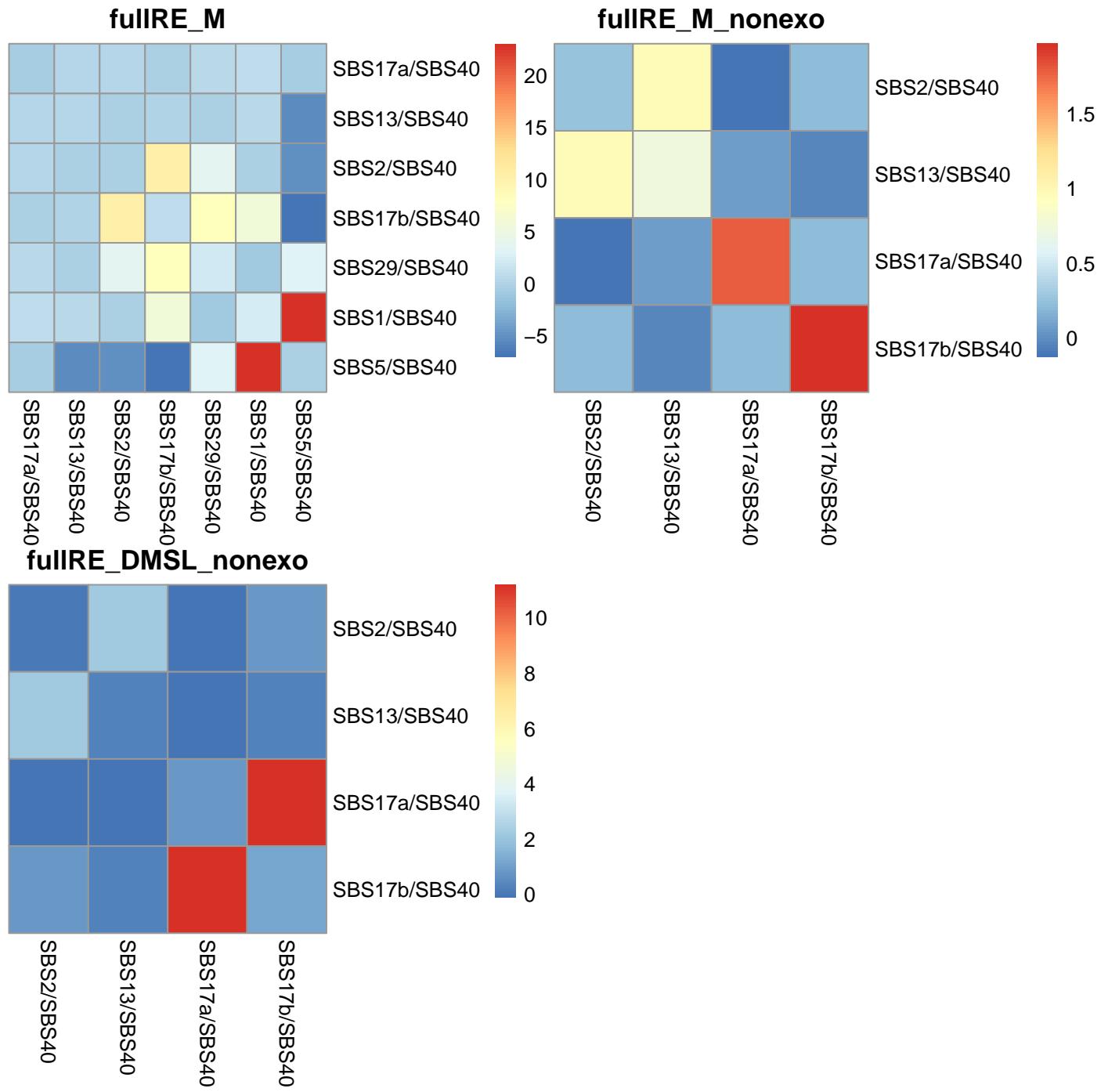
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 0.2664763.

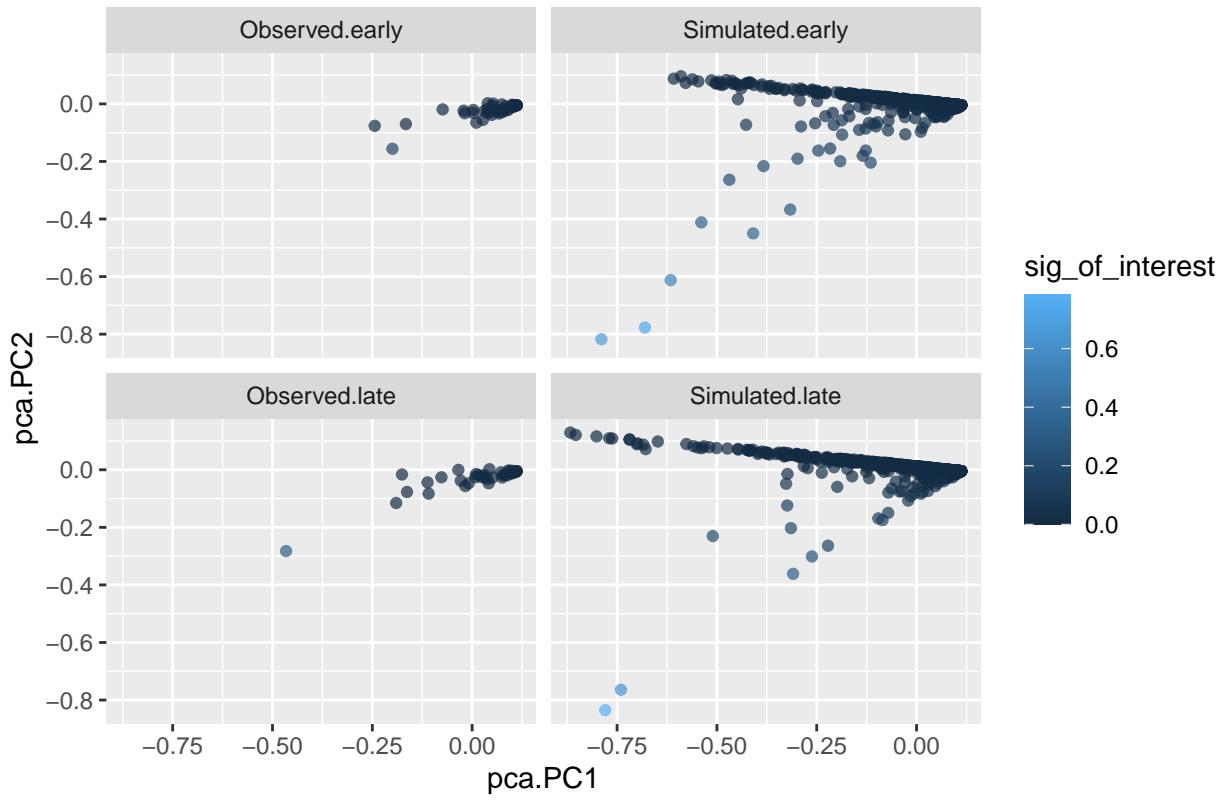
Covariance matrices



Simulation under inferred data

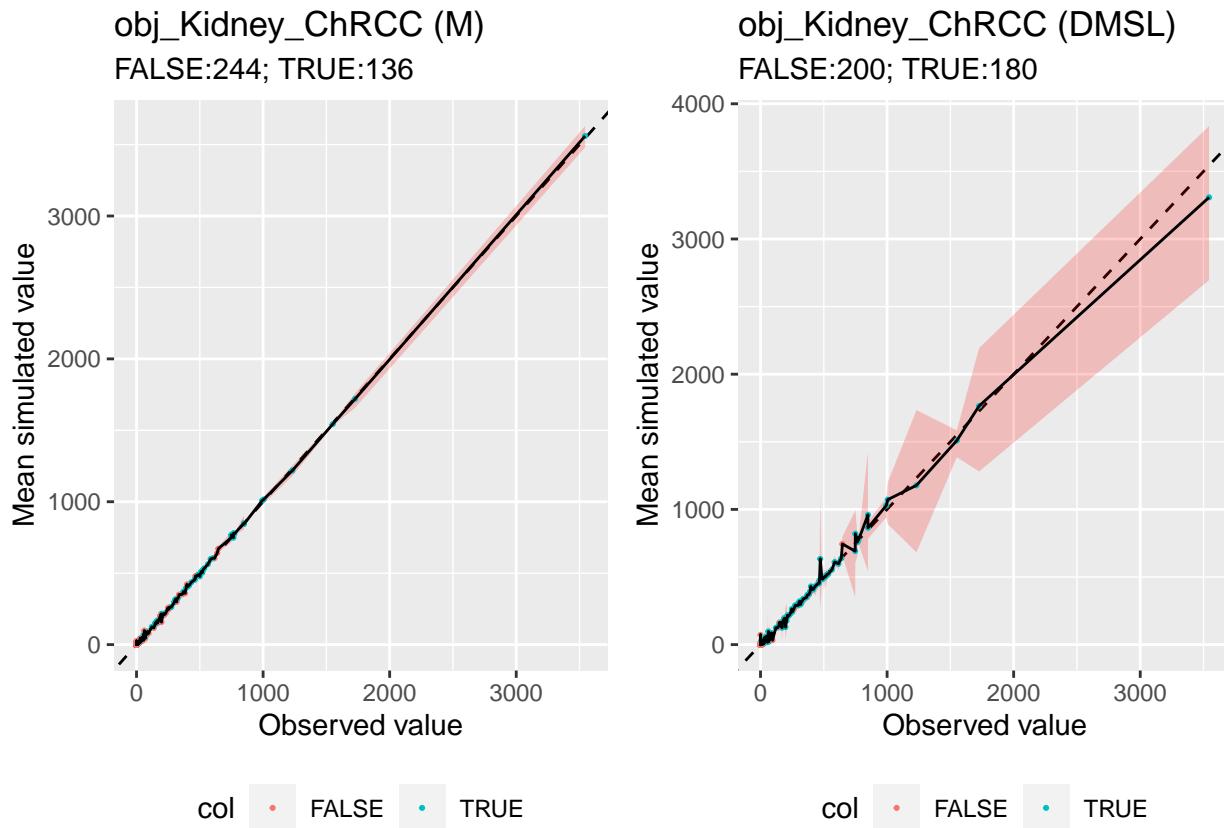
```
## [1] 38
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

Simulation of Kidney–ChRCC samples



Ranked plot for coverage

```
ct <- "Kidney-ChRCC"
integer_overdispersion_param_DMSL <- 1
obj_Kidney_ChRCC_nonexo <- give_subset_sigs_TMBobj(obj_Kidney_ChRCC, 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_Kidney_ChRCC_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_Kidney_ChRCC_nonexo,
loglog = F, title = 'obj_Kidney_ChRCC (M)'),
give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_DMSL_nonexo,
data_object = obj_Kidney_ChRCC_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_Kidney_ChRCC_nonexo,
loglog = F, title = 'obj_Kidney_ChRCC (DMSL)'), ncol=2)
```



Signatures from mutSigExtractor

The signatures from mutSigExtractor are as follows:

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

## [1] 38

give_barplot_from_obj(obj = obj_Kidney_ChRCC_mutSigExtractor, legend_on = FALSE)

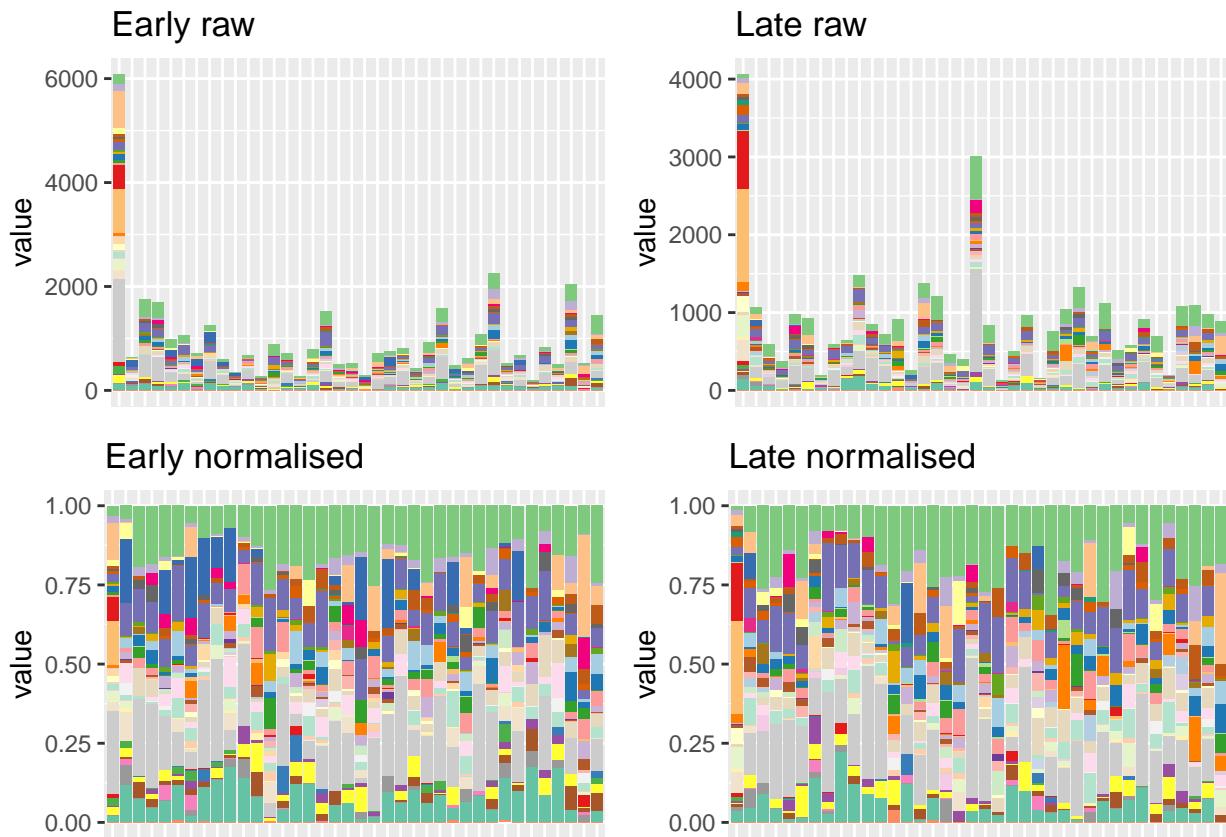
## Creating plot... it might take some time if the data are large. Number of samples: 38
## Creating plot... it might take some time if the data are large. Number of samples: 38
## Creating plot... it might take some time if the data are large. Number of samples: 38
## Creating plot... it might take some time if the data are large. Number of samples: 38

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

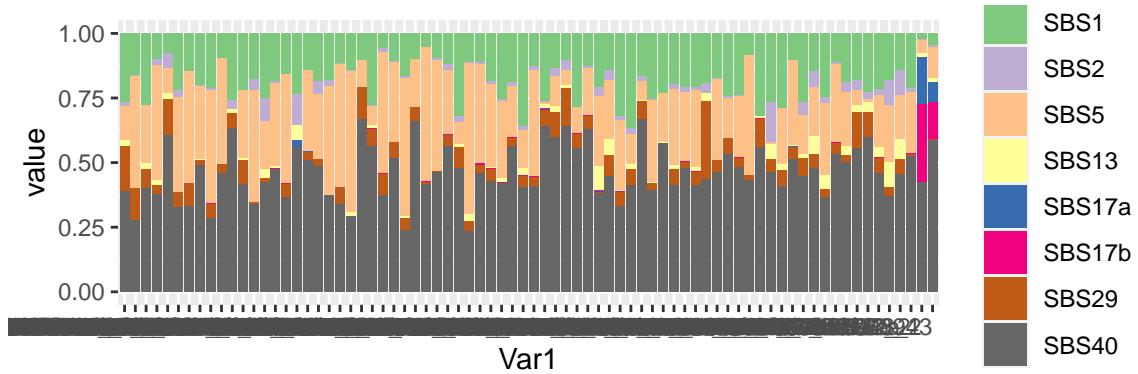
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



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_Kidney_ChRCC$Y)),
              order_labels = names(sort(rowSums(non_duplicated_rows(obj_Kidney_ChRCC$Y)),
                                         decreasing = F)))
```

Creating plot... it might take some time if the data are large. Number of samples: 76



Kidney-RCC.clearcell

Kidney-RCC.papillary

Liver-HCC

Lung-AdenoCA

Lung-SCC

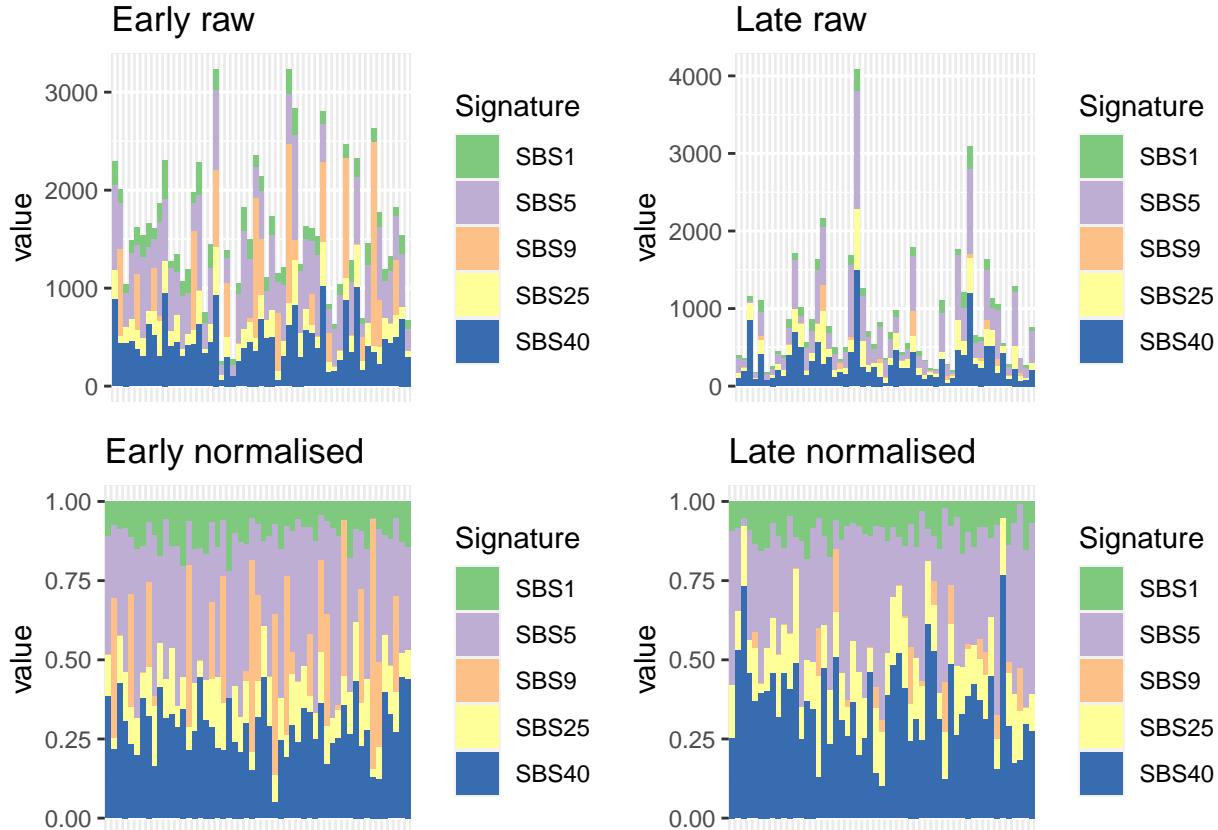
Lymph-BNHL

Lymph-CLL

Barplot and general statistics

```
## [1] 53
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 53
## Creating plot... it might take some time if the data are large. Number of samples: 53
## Creating plot... it might take some time if the data are large. Number of samples: 53
## Creating plot... it might take some time if the data are large. Number of samples: 53
```



The number of samples and signatures is:

```
## [1] 106 5
```

The signatures are:

```
## [1] "SBS1" "SBS5" "SBS9" "SBS25" "SBS40"
```

Convergence table

We have converged results in most cases

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

Potentially problematic signatures

SBS9 has quite a lot of zeros.

```
colSums(obj_Lymph CLL$Y == 0)/nrow(obj_Lymph CLL$Y)

##      SBS1      SBS5      SBS9      SBS25      SBS40
## 0.000000000 0.028301887 0.613207547 0.009433962 0.000000000

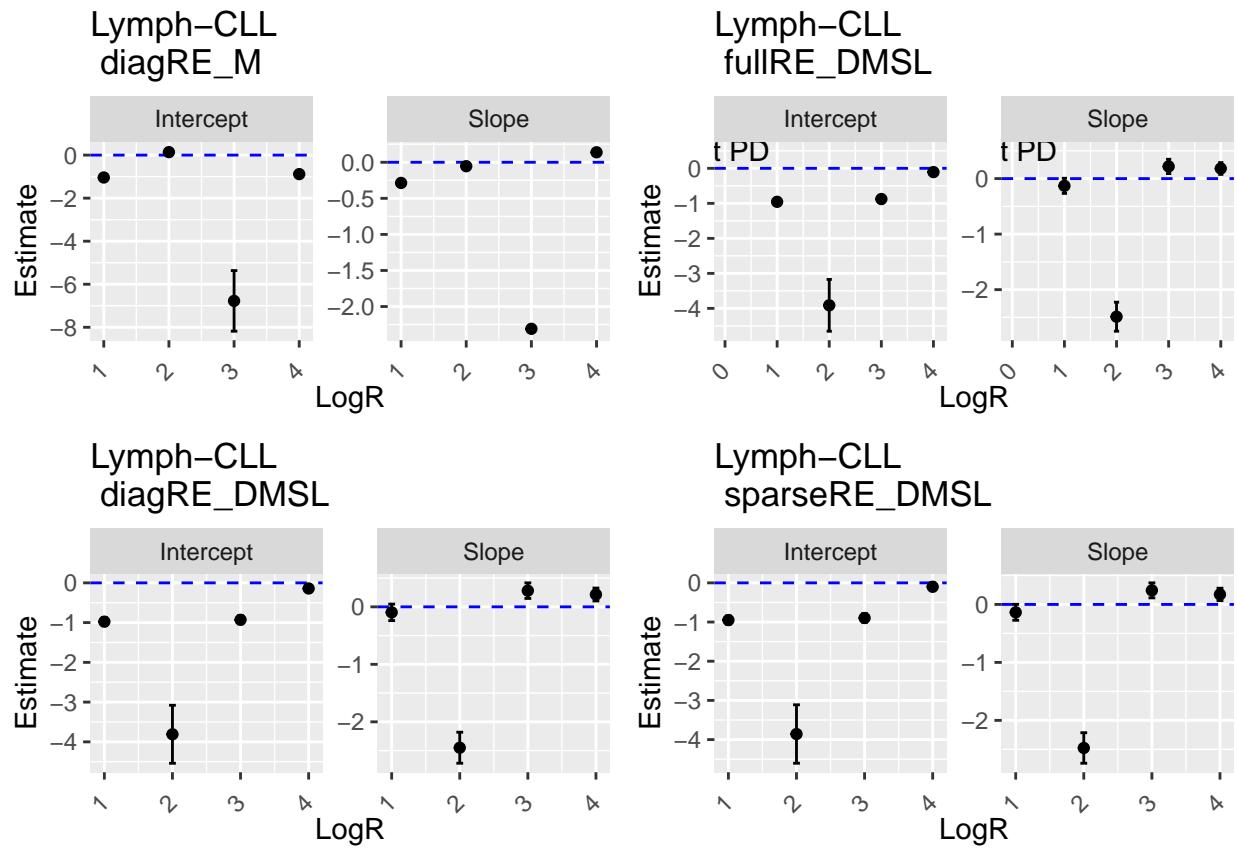
colSums(obj_Lymph CLL$Y)/sum(obj_Lymph CLL$Y)

##      SBS1      SBS5      SBS9      SBS25      SBS40
## 0.09712712 0.33726681 0.12275176 0.13805198 0.30480234
```

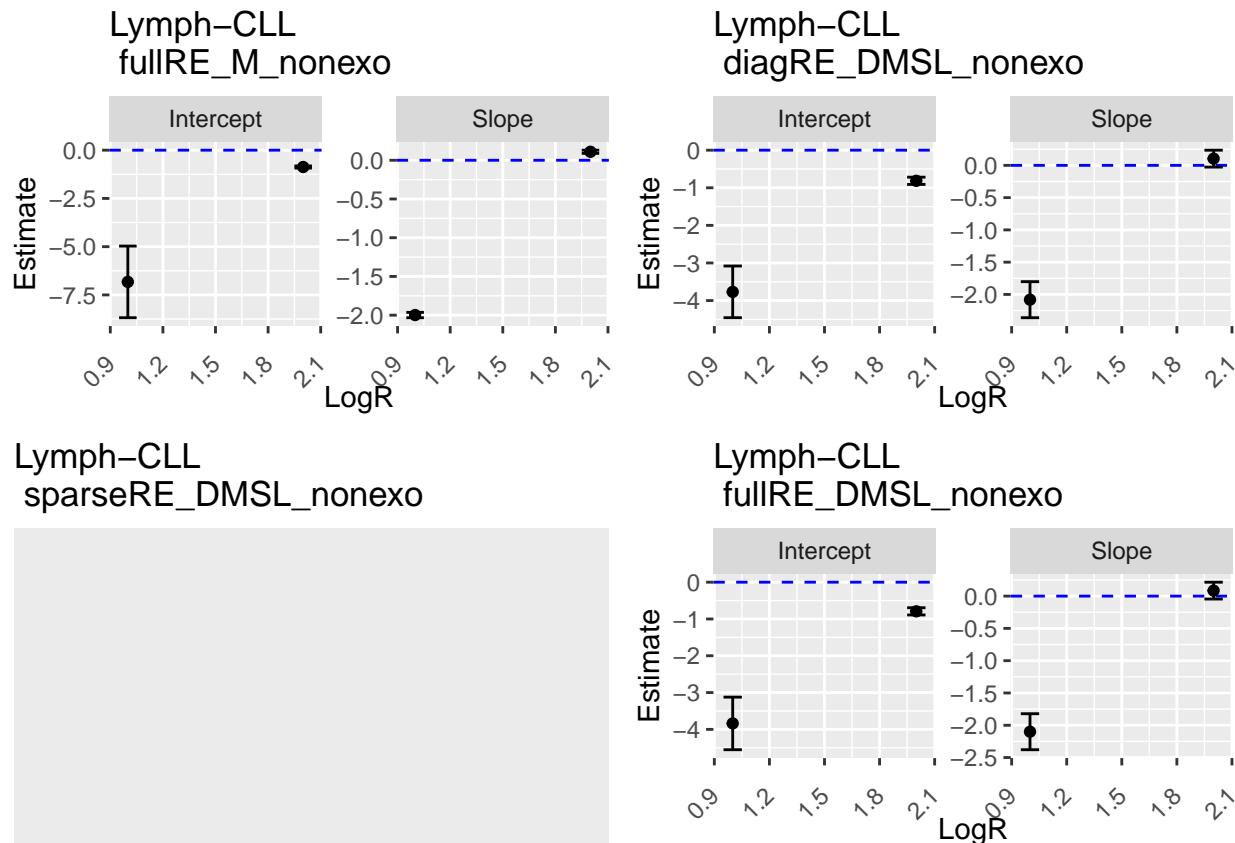
Betas

```
ct <- "Lymph-CLL"

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(diagRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n diagRE_DMSL_nonexo')),
  plot_betas(sparseRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n sparseRE_DMSL_nonexo')),
  plot_betas(fullRE_DMSL_nonexo[[ct]])+ggtitle(paste0(ct, '\n fullRE_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

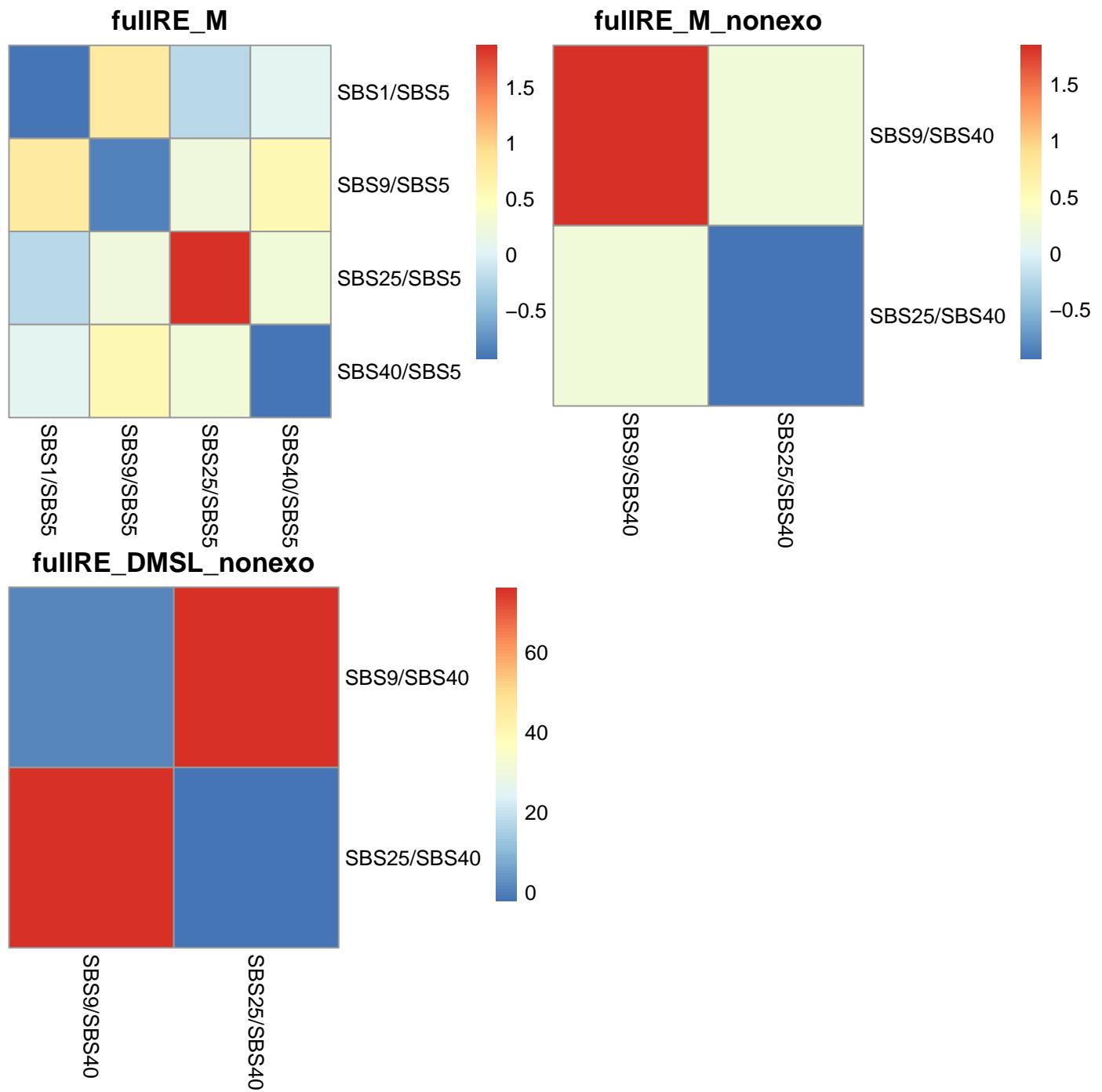
## Warning in if (is.na(idx_beta)) {: the condition has length > 1 and only the
## first element will be used

## 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 $6.1779312 \times 10^{-14}$.

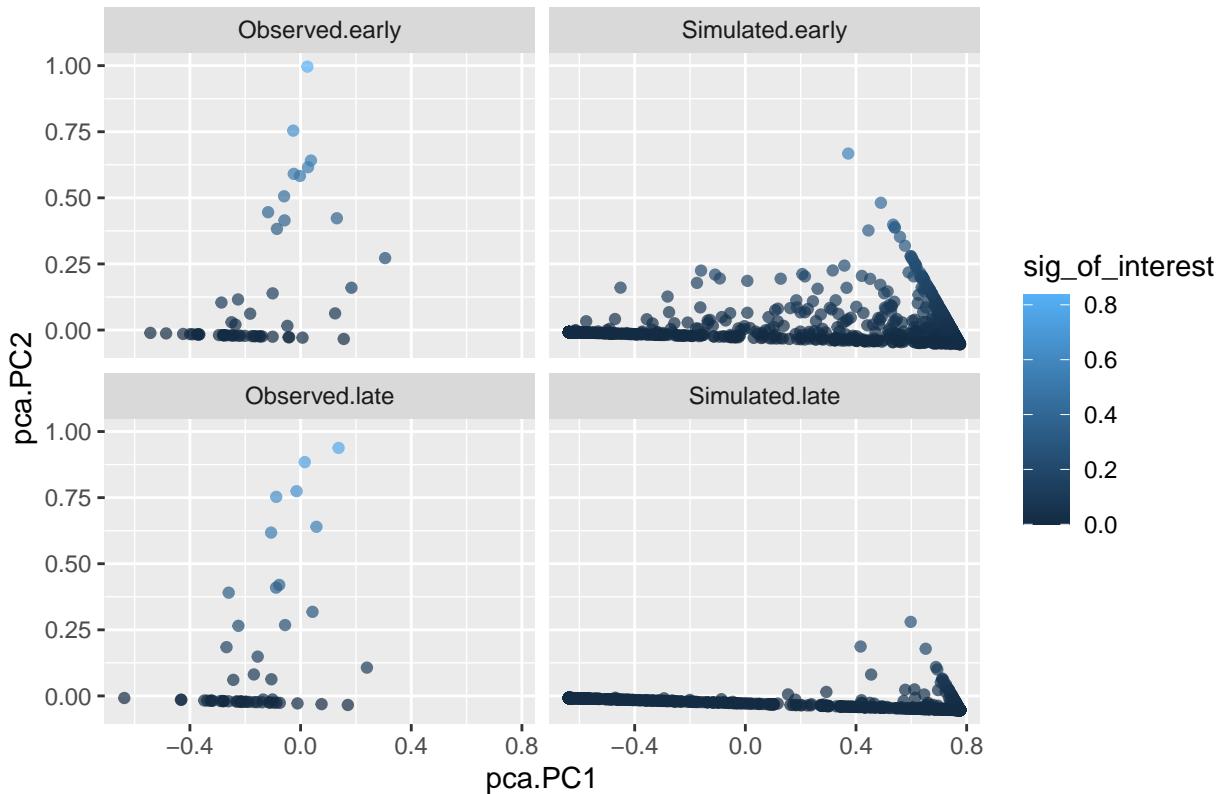
Covariance matrices



Simulation under inferred data

```
## [1] 53
## Warning in mvtnorm::rmvnorm(n = n_sim, mean = rep(0, dmin1), sigma = cov_mat):
## sigma is numerically not positive semidefinite
```

Simulation of Lymph-CLL samples



Ranked plot for coverage

```
ct <- "Lymph-CLL"
integer_overdispersion_param_DMSL <- 1
obj_Lymph_CLL_nonexo <- give_subset_sigs_TMBobj(obj_Lymph_CLL, sigs_to_remove = nonexogenous$V1)
for(loglog_bool_it in c(T,F)){
  .full_rankedplot <- give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_
    data_object = obj_Lymph_CLL_nonexo,
    print_plot = F, nreps = 100, 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_Lymph_CLL_nonexo,
    loglog = loglog_bool_it, title = 'obj_Lymph_CLL_nonexo (fullRE DMSL)')
  grid.arrange(
    give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_M_nonexo,
      data_object = obj_Lymph_CLL_nonexo,
      print_plot = F, nreps = 100, 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_Lymph_CLL_nonexo,
        loglog = loglog_bool_it, title = 'obj_Lymph_CLL_nonexo (M)'),
    .full_rankedplot, ncol=2)

  grid.arrange(give_interval_plots_2(df_rank = lapply(list(give_ranked_plot_simulation(tmb_fit_object = fullRE_M_nonexo,
    data_object = obj_Lymph_CLL_nonexo,
    print_plot = F, nreps = 100, 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_Lymph_CLL_nonexo,
      loglog = loglog_bool_it, title = 'obj_Lymph_CLL_nonexo (M)'),
```

```
    data_object = obj_Lymph CLL_nonexo,
    print_plot = F, nreps = 100, 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_Lymph CLL_nonexo,
    loglog = loglog_bool_it, title = 'obj_Lymph CLL nonexo (diagRE DMSL)'),
    .full_rankedplot, ncol=2)

}

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

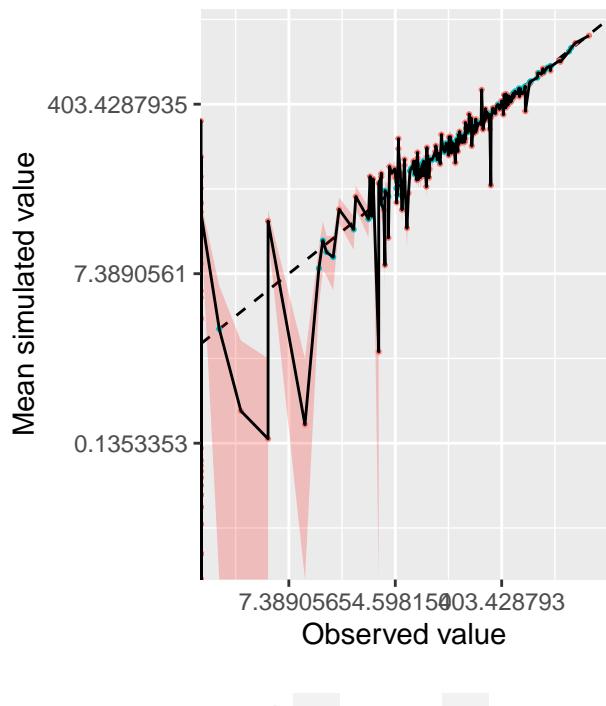
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

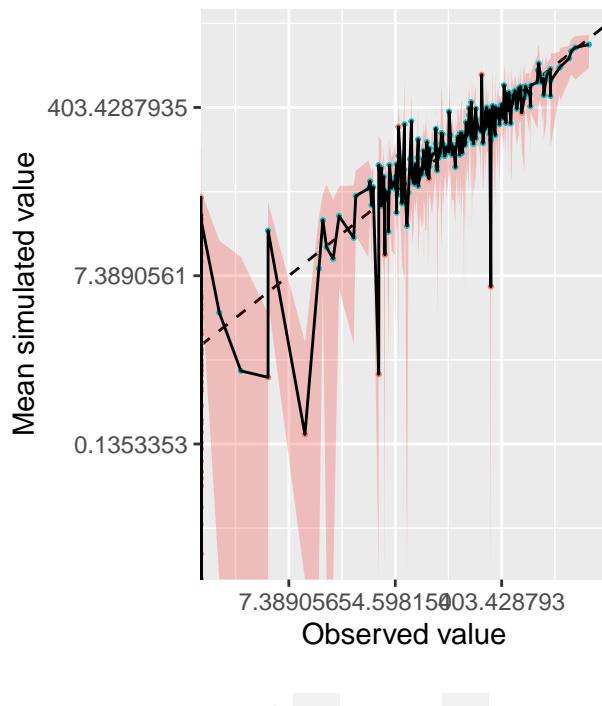
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
```

obj_Lymph CLL nonexo (M
FALSE:188; TRUE:130



obj_Lymph CLL nonexo (fu
FALSE:75; TRUE:243



```
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis

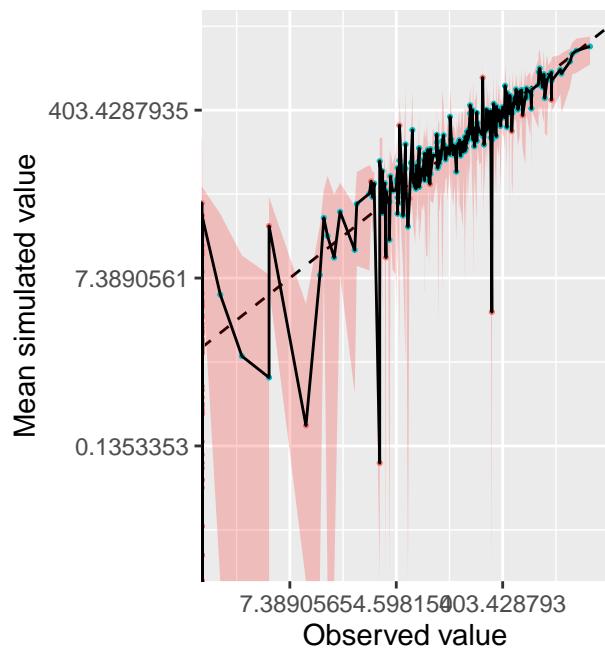
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
```

```

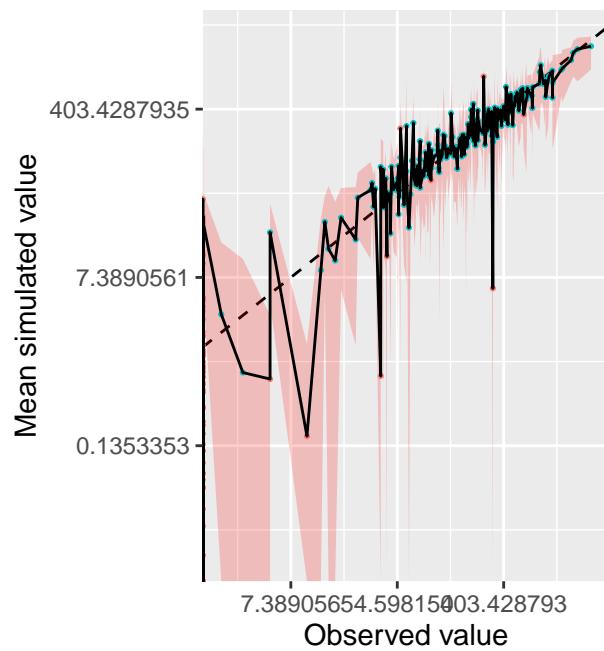
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis

```

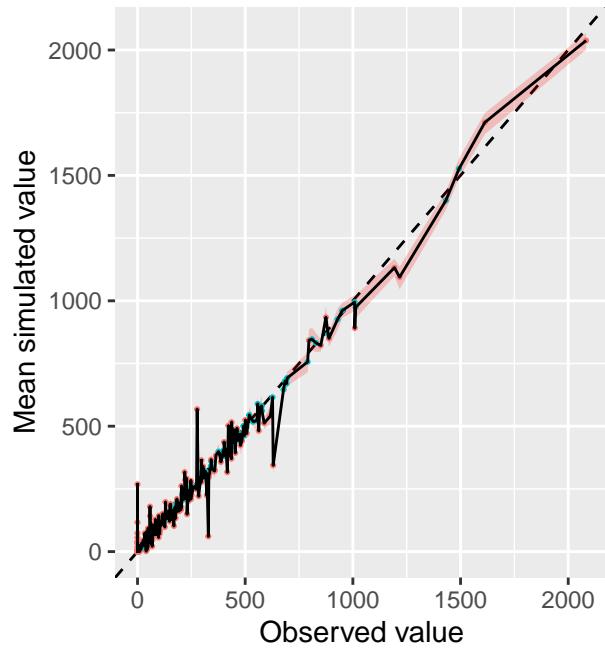
obj_Lymph CLL nonexo (di)
FALSE:79; TRUE:239



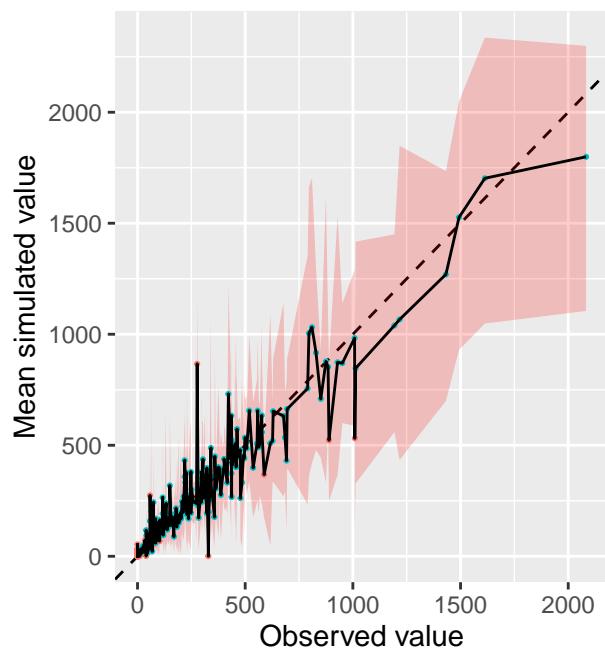
obj_Lymph CLL nonexo (fu)
FALSE:75; TRUE:243



obj_Lymph CLL nonexo (M)
FALSE:191; TRUE:127

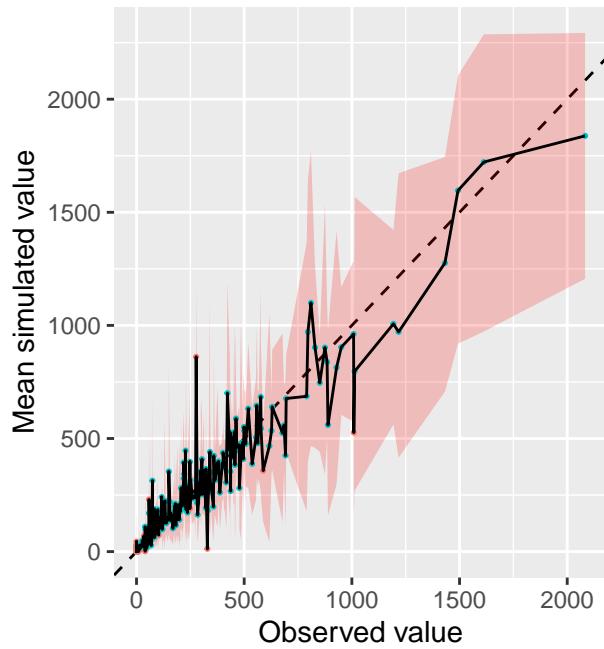


col FALSE TRUE
obj_Lymph CLL nonexo (diagRE)
FALSE:79; TRUE:239

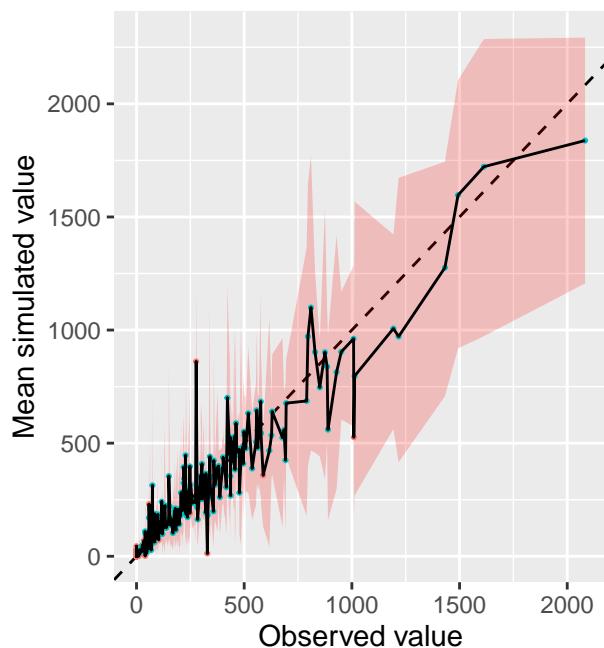


col FALSE TRUE

obj_Lymph CLL nonexo (fullRE L)
FALSE:78; TRUE:240



col FALSE TRUE
obj_Lymph CLL nonexo (fullRE L)
FALSE:78; TRUE:240



Signatures from mutSigExtractor

These are the signatures from mutSigExtractor:

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

```
## [1] 53
```

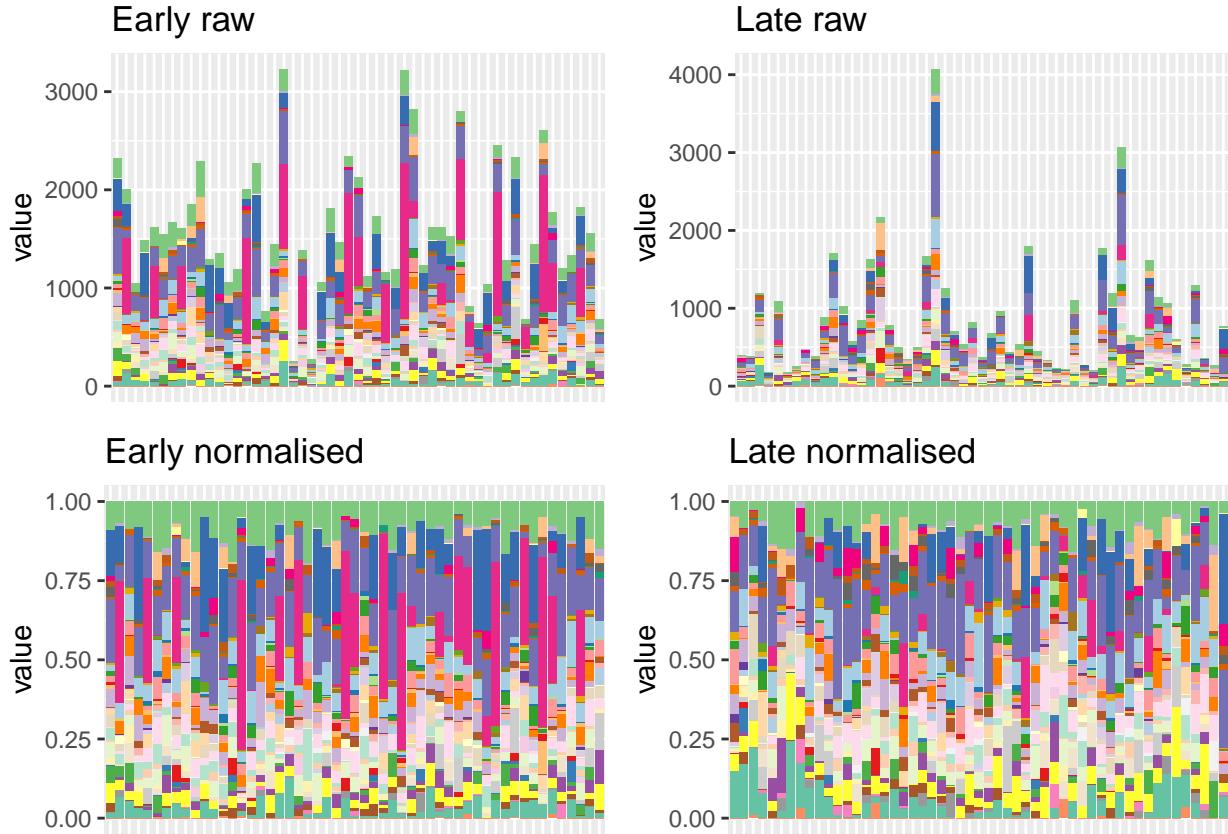
```
give_barplot_from_obj(obj = obj_Lymph CLL_mutSigExtractor, legend_on = FALSE)
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 53
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 53
```

```
## Creating plot... it might take some time if the data are large. Number of samples: 53
```

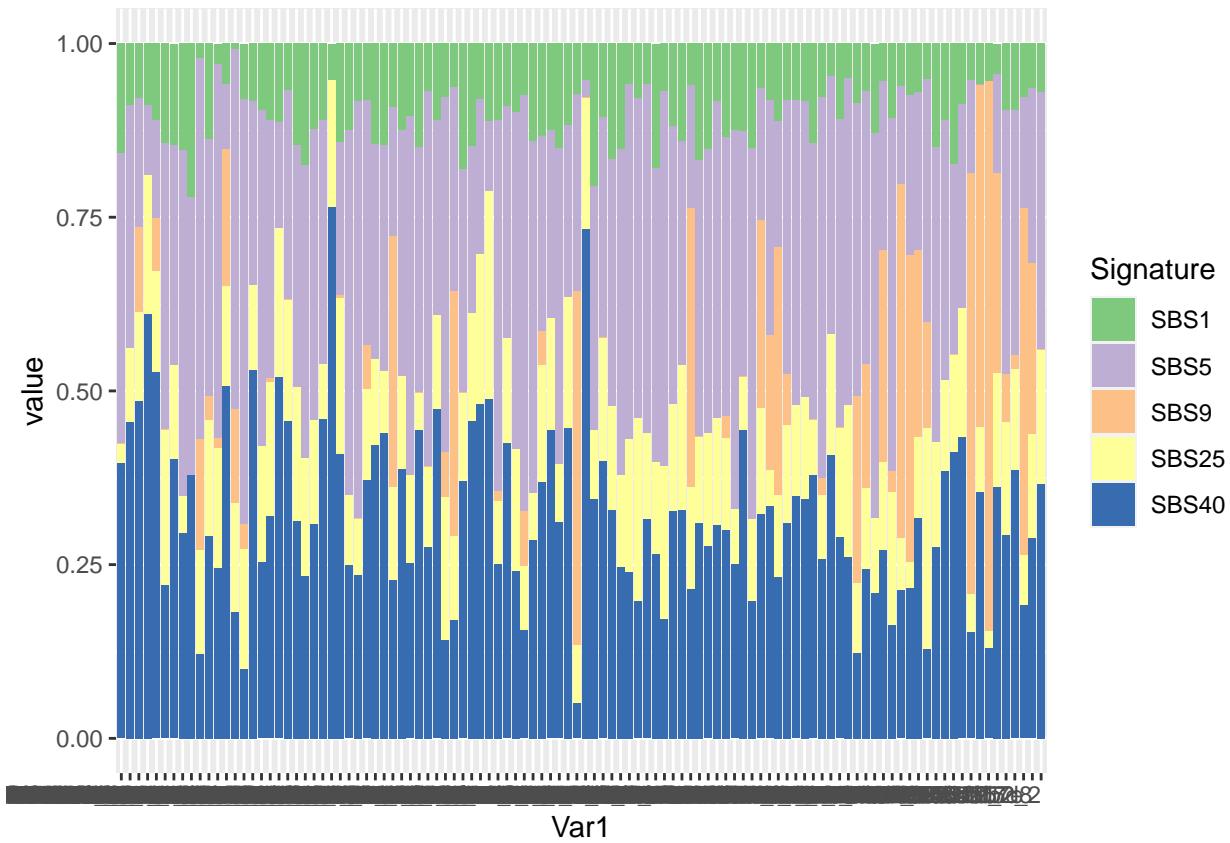
```
## Creating plot... it might take some time if the data are large. Number of samples: 53
```



Exposures sorted by increasing number of mutations: SBS9 and SBS25 seem to be somewhat associated with samples with a high number of mutations.

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

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



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 % Tue Jun 1 20:41:50 2021

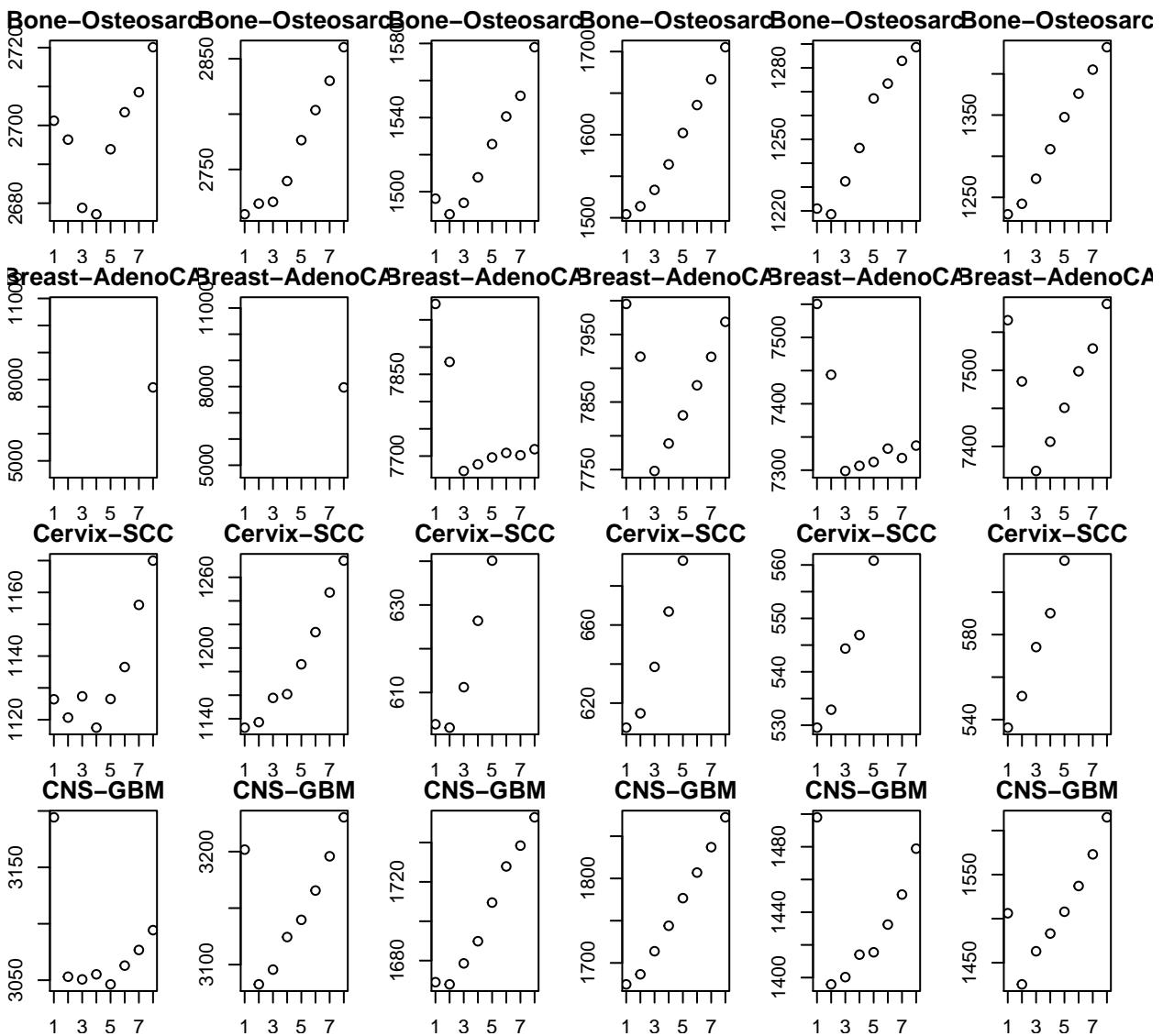
	ct	pvalue	model
1	Bone-Osteosarc	0.00	diagRE_DMSL_nonexo
2	Breast-AdenoCA	0.00	diagRE_DMSL_nonexo
3	Cervix-SCC	0.00	fullRE_DMSL_nonexo
4	CNS-Oligo	0.52	fullRE_DMSL_nonexo

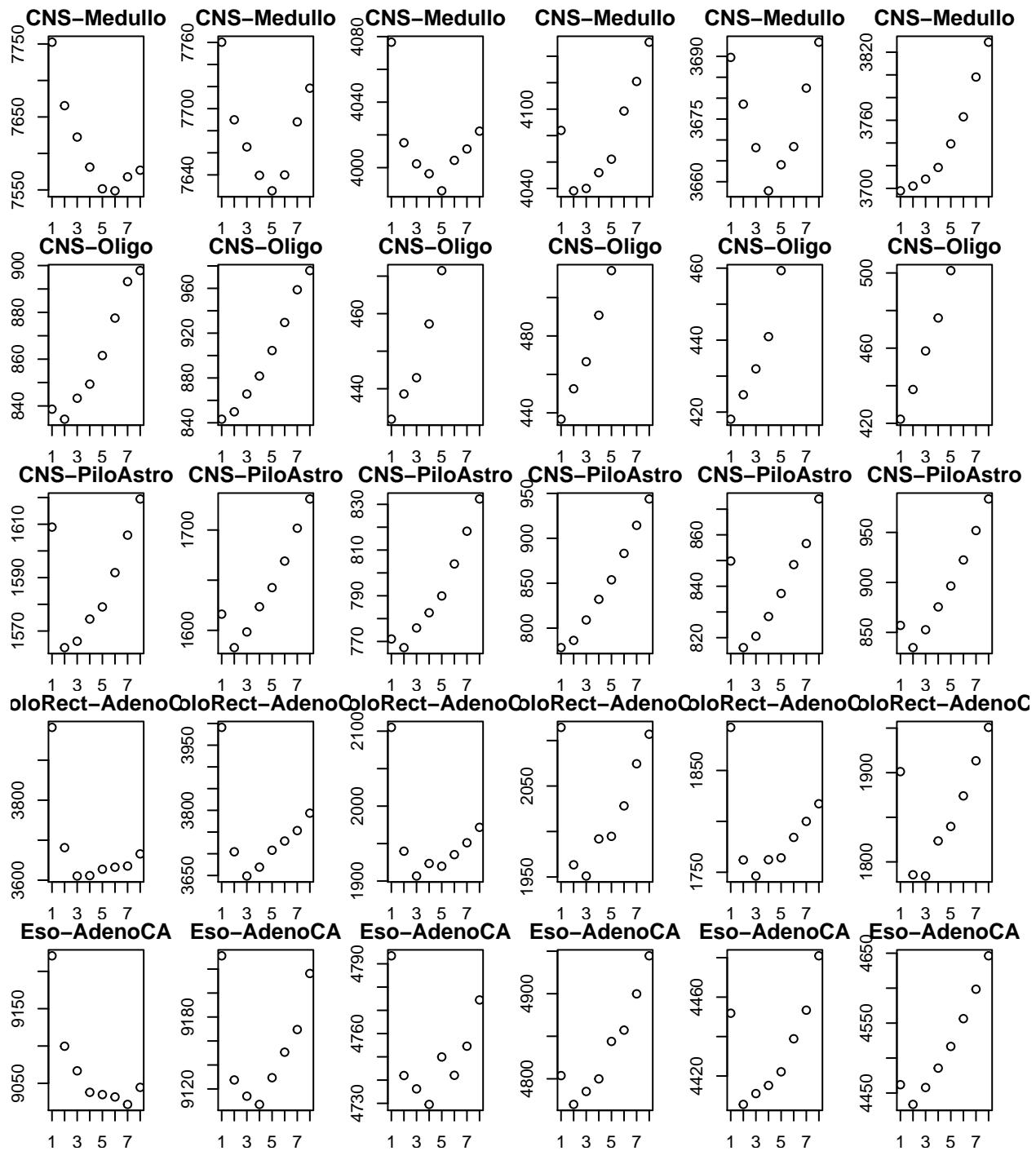
Dirichlet-Multinomial Mixtures

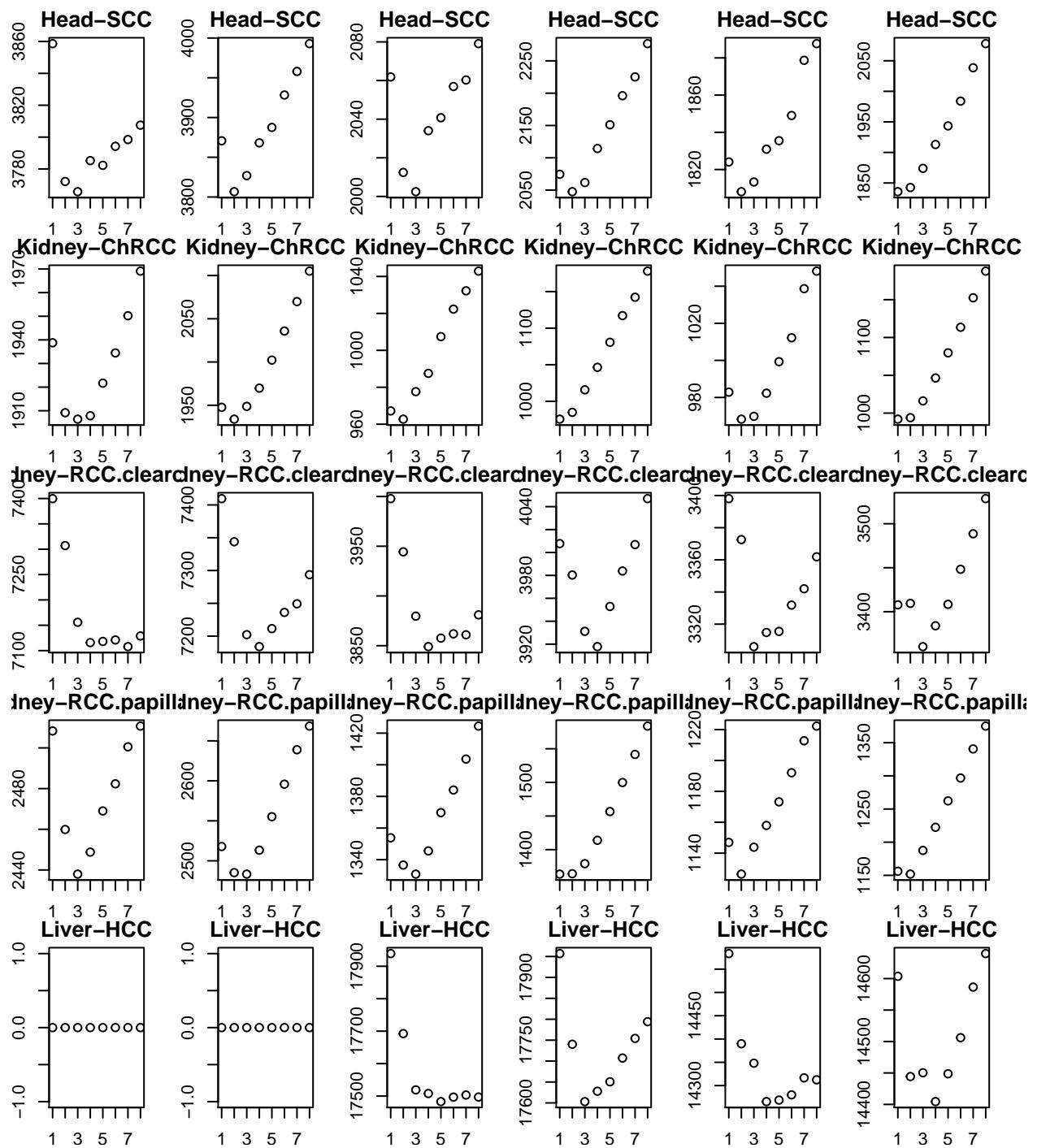
We run the software MicrobeDMMv1.0 to determine whether we are facing DMM mixtures or not.

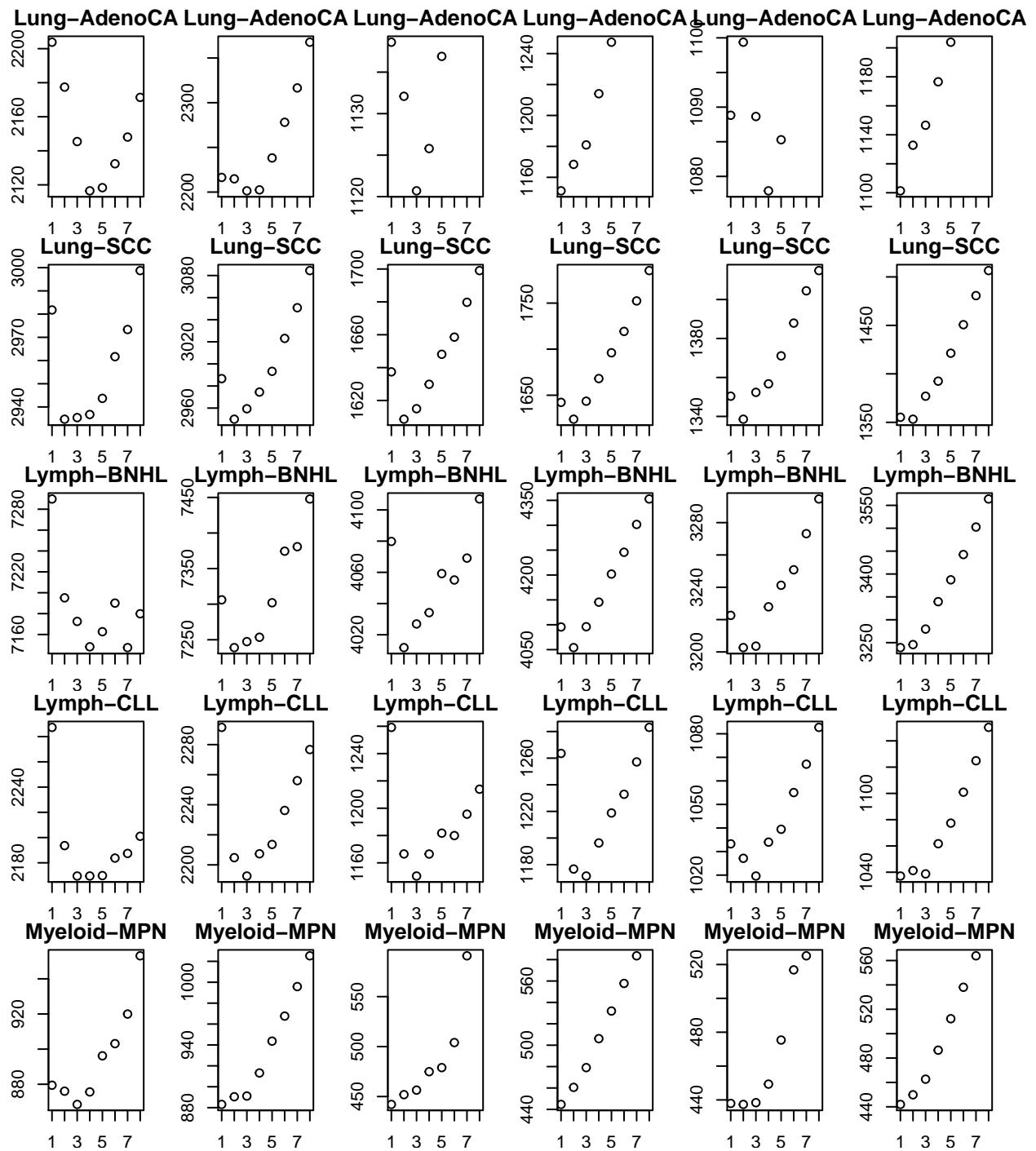
We save the files in two ways: all of the samples - early or not - together, and separately.

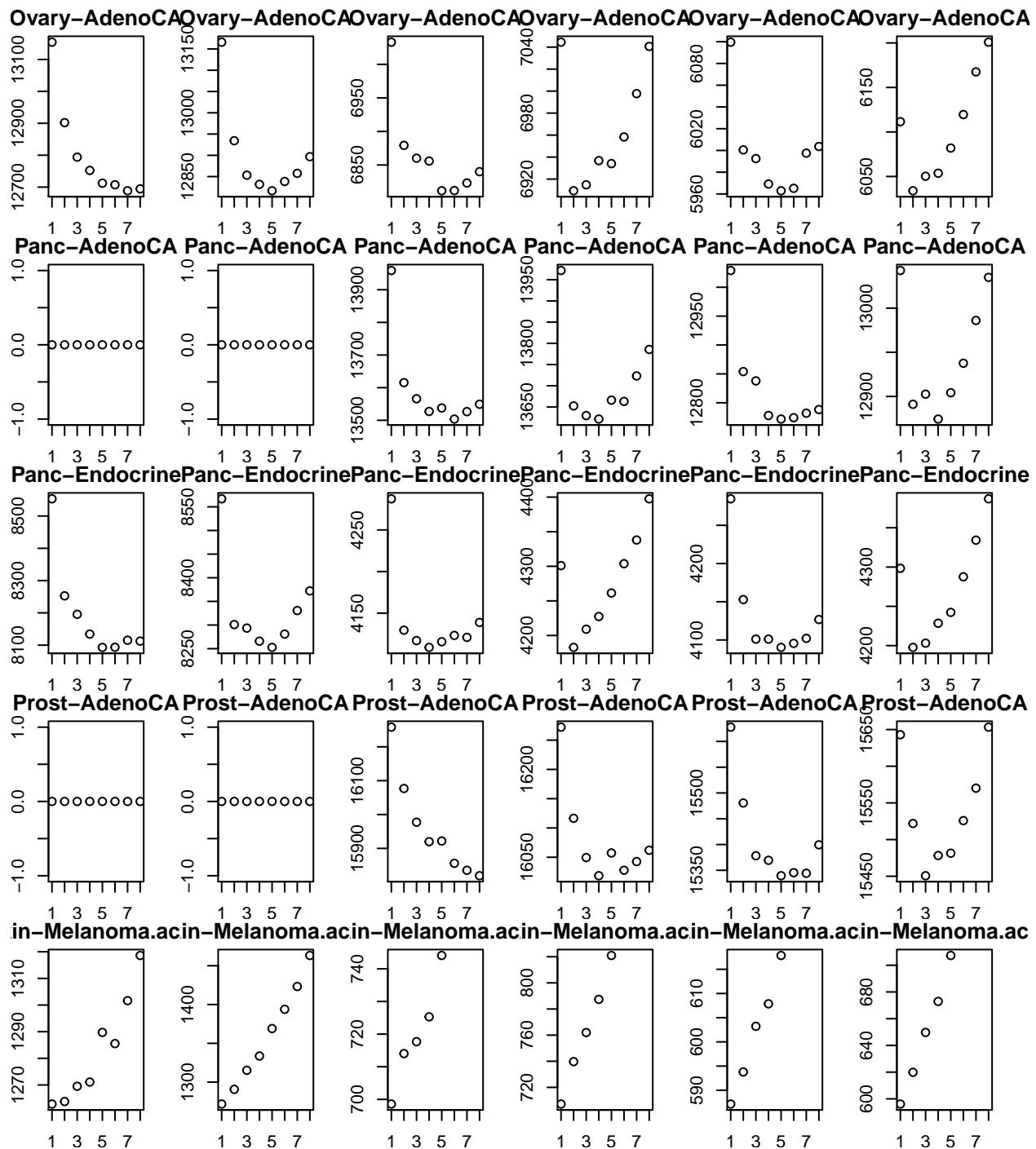
In some cases DMM says that there is an error with the input file - in this case the AIC or BIC is not plotted. If all of them are missing, all BIC and AIC are set to zero.

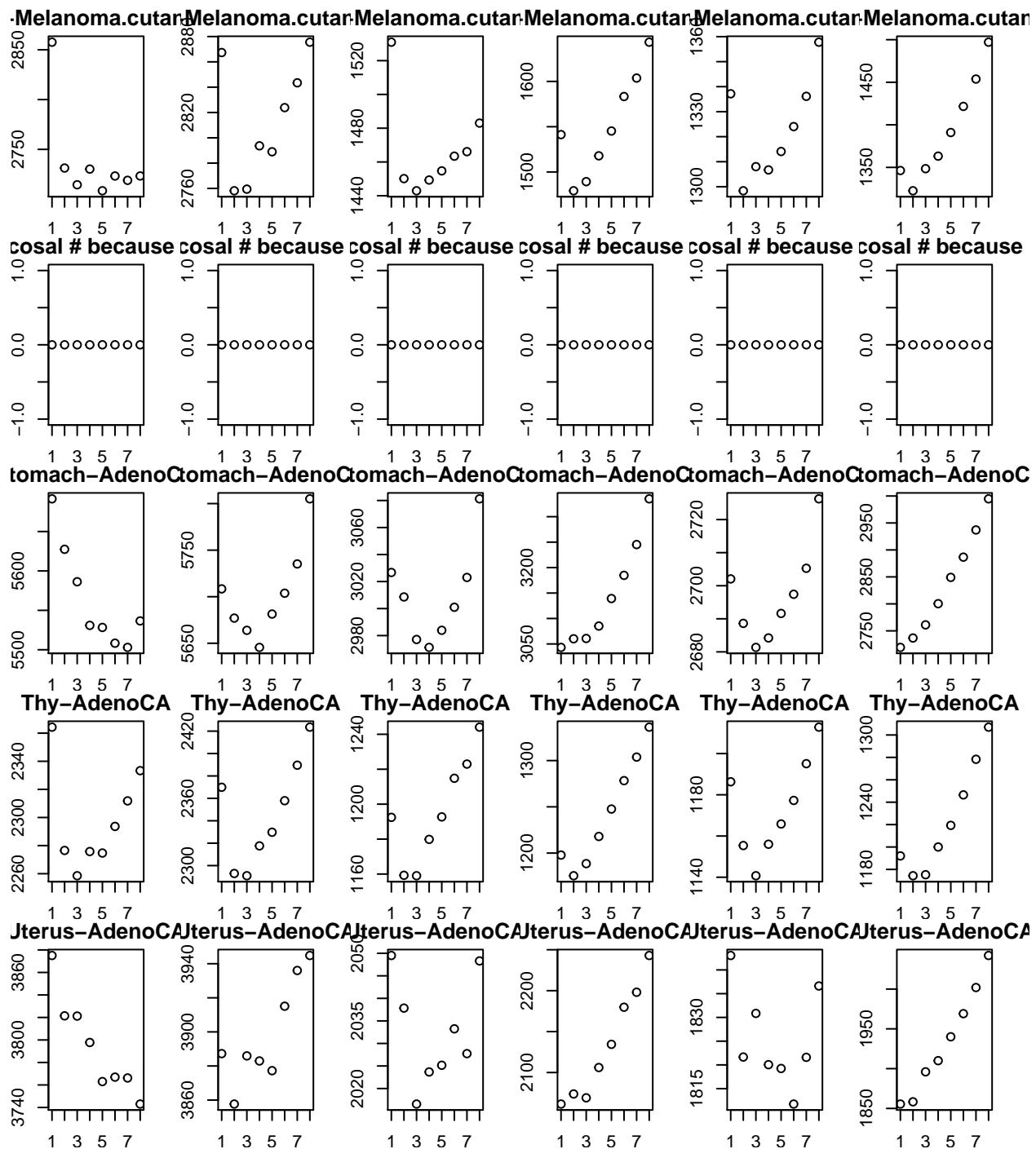












Comparison of signature exposures with QP and mutsigextractor

```
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)))

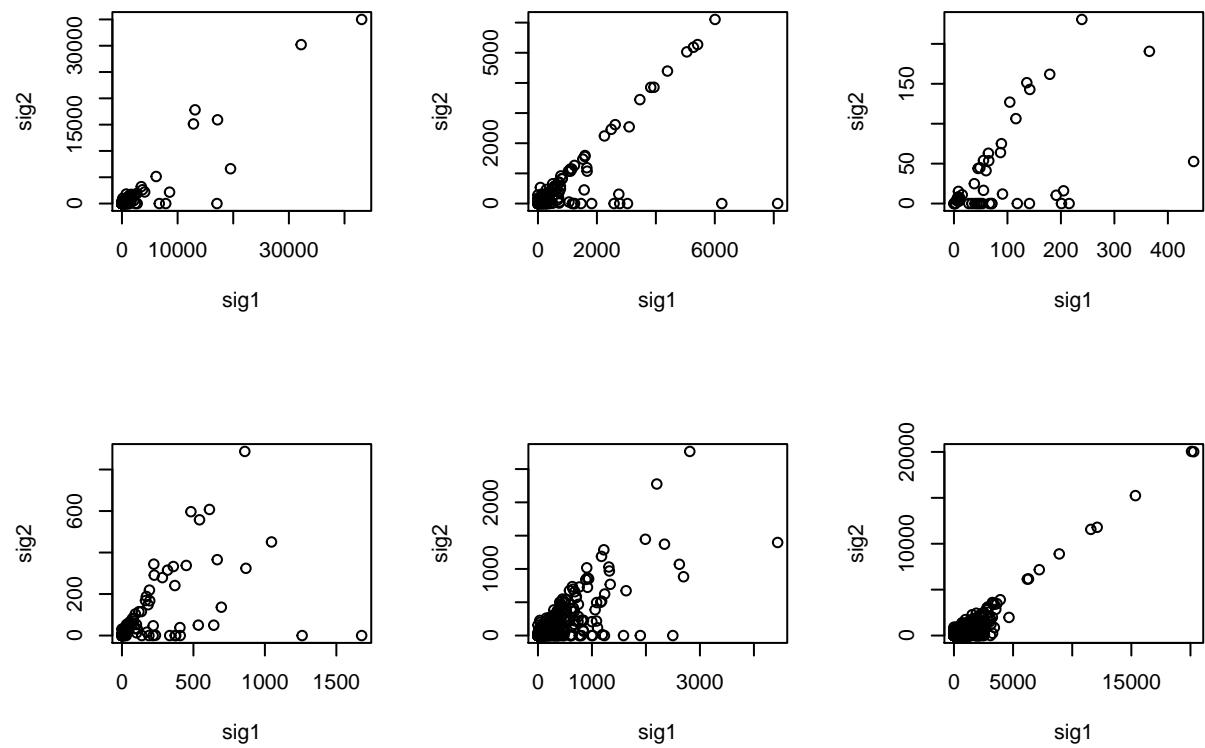
par(mfrow=c(2,3))
for(i in names(roo_obj2)){
  try({

    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

cbind.data.frame(pvals_fullRE_M=pvals_fullRE_M,
                  pvals_diagRE_DM=pvals_diagRE_DM,
                  pvals_DM=pvals_DM,
                  pvals_DMnonexo=pvals_DMnonexo)

##                                     pvals_fullRE_M pvals_diagRE_DM      pvals_DM
## Bone-Osteosarc          0.000000e+00  2.847099e-08 1.283415e-08
## Breast-AdenoCA          0.000000e+00  2.366485e-35 3.192087e-34
## Cervix-SCC              1.269057e-204 1.109283e-01 1.172648e-01
## CNS-GBM                 0.000000e+00  9.413763e-09 3.147710e-02
## CNS-Medullo            2.636002e-129 1.751080e-02 1.621432e-02
## CNS-Oligo                2.055546e-155 3.429210e-02 3.147710e-02
## CNS-PiloAstro           8.624590e-48   3.429210e-02 3.352191e-02
## ColoRect-AdenoCA        0.000000e+00  6.922344e-23 2.607337e-22
## Eso-AdenoCA             0.000000e+00  1.959292e-19 1.616523e-20
## Head-SCC                0.000000e+00  1.812725e-04 6.168366e-05
## Kidney-ChRCC            0.000000e+00  3.298283e-06 2.968403e-06

```

## Kidney-RCC.clearcell	0.000000e+00	1.236684e-24	6.365567e-25
## Kidney-RCC.papillary	0.000000e+00	5.703174e-18	2.271530e-18
## Liver-HCC	0.000000e+00	2.452590e-65	6.112670e-66
## Lung-AdenoCA	0.000000e+00	1.352595e-02	4.666292e-05
## Lung-SCC	0.000000e+00	1.523581e-20	1.688513e-22
## Lymph-BNHL	0.000000e+00	4.795960e-12	2.058732e-13
## Lymph-CLL	0.000000e+00	1.534926e-19	3.305201e-21
## Myeloid-MPN	4.472448e-113	9.320422e-08	4.518344e-08
## Ovary-AdenoCA	0.000000e+00	1.150681e-21	7.691981e-27
## Panc-AdenoCA	0.000000e+00	2.826716e-72	2.622351e-77
## Panc-Endocrine	0.000000e+00	2.784969e-18	8.995411e-20
## Prost-AdenoCA	0.000000e+00	1.603706e-88	3.647368e-91
## Skin-Melanoma.acral	5.621465e-210	1.802106e-01	1.475018e-01
## Skin-Melanoma.cutaneous	0.000000e+00	8.452268e-20	3.079994e-22
## Stomach-AdenoCA	0.000000e+00	9.504291e-04	1.166767e-05
## Thy-AdenoCA	6.170318e-310	2.707976e-02	1.621432e-02
## Uterus-AdenoCA	0.000000e+00	3.298283e-06	1.091244e-07
## pvals_DMnonexo			
## Bone-Osteosarc	NA		
## Breast-AdenoCA	3.347067e-11		
## Cervix-SCC	1.023813e-02		
## CNS-GBM	8.425463e-04		
## CNS-Medullo	4.674452e-01		
## CNS-Oligo	NA		
## CNS-PiloAstro	6.382416e-01		
## ColoRect-AdenoCA	4.799950e-15		
## Eso-AdenoCA	2.352755e-19		
## Head-SCC	1.221425e-03		
## Kidney-ChRCC	6.382416e-01		
## Kidney-RCC.clearcell	7.826378e-20		
## Kidney-RCC.papillary	3.446472e-06		
## Liver-HCC	7.946301e-50		
## Lung-AdenoCA	1.023813e-02		
## Lung-SCC	1.669362e-01		
## Lymph-BNHL	1.412634e-05		
## Lymph-CLL	NA		
## Myeloid-MPN	NA		
## Ovary-AdenoCA	4.790258e-20		
## Panc-AdenoCA	1.789201e-44		
## Panc-Endocrine	5.008201e-08		
## Prost-AdenoCA	3.131201e-61		
## Skin-Melanoma.acral	NA		
## Skin-Melanoma.cutaneous	NA		
## Stomach-AdenoCA	7.963662e-02		
## Thy-AdenoCA	4.674452e-01		
## Uterus-AdenoCA	8.425463e-04		