

# 2023 554 SUMMER Package R Notes

Jon Wakefield  
Departments of Biostatistics and Statistics  
University of Washington

2023-01-24

## Small Area Estimation (SAE)

In these notes, SAE via the `SUMMER` package will be illustrated.

Details on `SUMMER`, including a vignette, can be found [here](#)

We illustrate with the Washington State BRFSS diabetes example and will obtain:

- Naive estimates
- Weighted estimates
- Estimates from a binomial BYM2 model
- Estimates from Fay-Herriot models

## Load `SUMMER` package

We first load the `SUMMER` package.

INLA is not in a standard repository, so we check if it is available and install it if it is not.

```
library(SUMMER)
if (!isTRUE(requireNamespace("INLA", quietly = TRUE))) {
  install.packages("INLA", repos=c(getOption("repos"), INLA="https://inla.r-inla-download.org/R/stable"),
  }
```

## Read in Data

BRFSS contains the full BRFSS dataset with 16,283 observations:

- `diab2` variable is the binary indicator of Type II diabetes
- `strata` is the strata indicator and
- `rwt_11cp` is the final weight.

For the purpose of this analysis, we first remove records with missing HRA code or diabetes status from this dataset.

```
data(BRFSS)
BRFSS <- subset(BRFSS, !is.na(BRFSS$diab2))
BRFSS <- subset(BRFSS, !is.na(BRFSS$hracode))
```

KingCounty contains the map of the King County HRAs. In order to fit spatial smoothing model, we first need to compute the adjacency matrix for the HRAs, `mat`, and make sure both the column and row names correspond to the HRA names.

```
library(sf) # Load sf for spatial analysis
library(prioritizr) # Allows us to create an adjacency matrix

data(KingCounty)
KingCounty <- st_as_sf(KingCounty)
nb.r <- st_centroid(st_geometry(KingCounty), of_largest_polygon=TRUE)
mat <- adjacency_matrix(KingCounty)
colnames(mat) <- rownames(mat) <- KingCounty$HRA2010v2_
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
mat[1:2,1:2]
##           Auburn-North Auburn-South
## Auburn-North           0           1
## Auburn-South           1           0
```

## Create survey object

We load the survey package and then define the survey object for the BRFSS data. We have stratified, disproportionate sampling, so note the arguments:

- `weights`
- `strata`

We then calculate the direct (weighted) estimates.

```
library(survey)
design <- svydesign(ids = ~1, weights = ~rwt_llcp,
                  strata = ~strata, data = BRFSS)
direct <- svyby(~diab2, ~hracode, design, svymean)
head(direct,n=7)
##           hracode      diab2      se
## Auburn-North      Auburn-North 0.10403154 0.02147752
## Auburn-South      Auburn-South 0.23293289 0.04897800
## Ballard           Ballard 0.07047572 0.02225241
## Beacon/Gtown/S.Park Beacon/Gtown/S.Park 0.08083033 0.02603522
## Bear Creek/Carnation/Duvall Bear Creek/Carnation/Duvall 0.05166773 0.01190146
## Bellevue-Central    Bellevue-Central 0.05914082 0.01485885
## Bellevue-NE         Bellevue-NE 0.05772789 0.01509705
```

## Binomial spatial smoothing model

We ignore the design and fit the model:

$$y_i | p_i \sim \text{Binomial}(n_i, p_i)$$

$$\theta_i = \log\left(\frac{p_i}{1 - p_i}\right) = \alpha + b_i$$

with  $b_i$  following a BYM2 model, i.e., an iid normal random effect and an intrinsic CAR (ICAR) random effect.

The binomial smoothing model is fitted by specifying NULL for the survey characteristics.

## The smoothSurvey function

Note how the polygon information is input, and the neighbors in the Amat argument - this is required for the ICAR.

```
smoothed <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = mat, responseType = "binary",
responseVar = "diab2", strataVar = NULL, weightVar = NULL, regionVar = "hrcode",
clusterVar = NULL, CI = 0.95)
```

The usual INLA summaries can be found in smoothed\$fit:

```
smoothed$fit$summary.fixed
##              mean          sd 0.025quant  0.5quant 0.975quant mode
## (Intercept) -2.352745  0.0308999  -2.414049 -2.352545 -2.292598  NA
##              kld
## (Intercept) 7.037843e-07
smoothed$fit$summary.hyper
##              mean          sd 0.025quant  0.5quant
## Precision for region.struct 15.1805533 4.9551884  7.7565428 14.4141749
## Phi for region.struct       0.8531176 0.1391289  0.4841199 0.8981503
##              0.975quant mode
## Precision for region.struct 27.0536113  NA
## Phi for region.struct       0.9952539  NA
```

Now examine some of the other components:

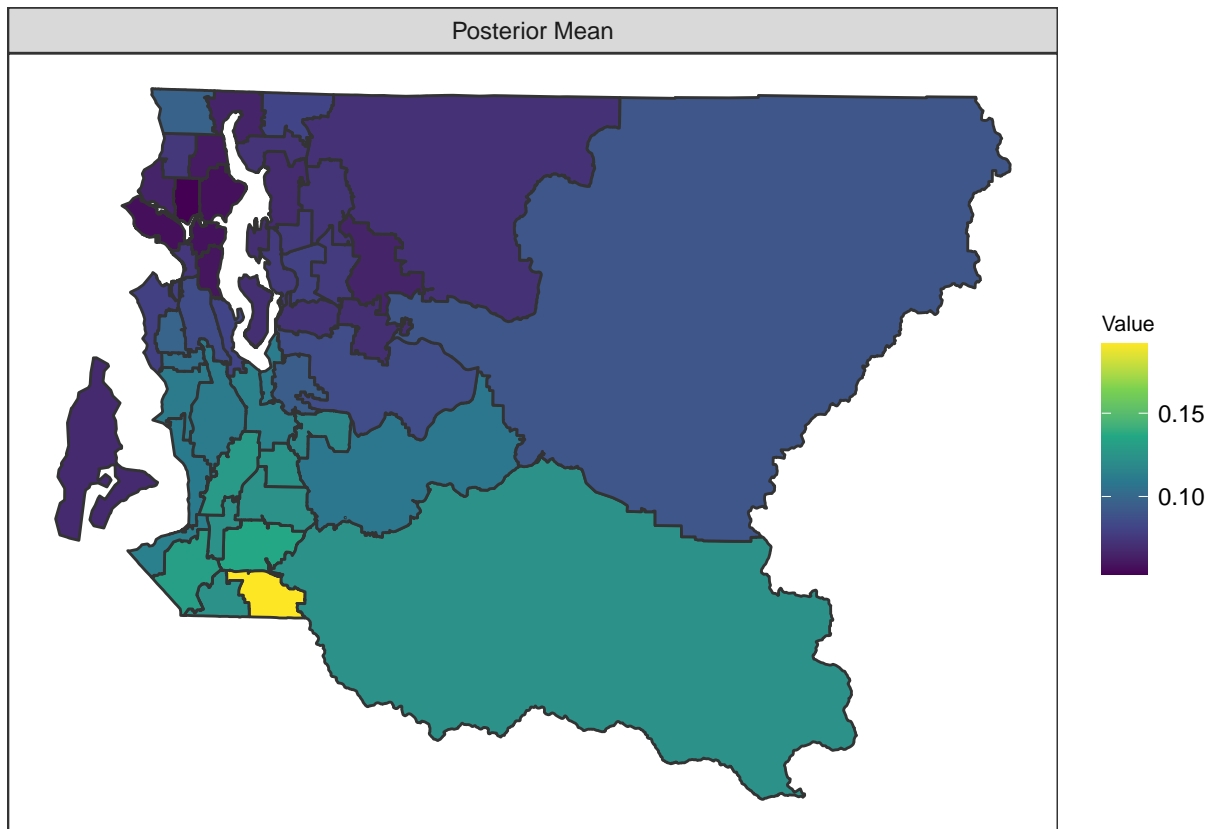
```
names(smoothed)
## [1] "HT"           "smooth"       "smooth.overall" "fit"
## [5] "CI"           "Amat"         "responseType"  "formula"
## [9] "msg"
names(smoothed$HT)
## [1] "region"       "HT.est"       "HT.var"       "HT.logit.est"
## [5] "HT.logit.var" "HT.logit.prec" "n"            "y"
names(smoothed$smooth)
## [1] "region"       "mean"         "var"          "median"       "lower"
## [6] "upper"        "logit.mean"   "logit.var"     "logit.median" "logit.lower"
## [11] "logit.upper"
head(smoothed$HT, n=4)
##              region      HT.est      HT.var HT.logit.est HT.logit.var
## 1      Auburn-North 0.14028777 0.0004338385  -1.812902  0.02982513
## 2      Auburn-South 0.23204420 0.0009845287  -1.196804  0.03100377
## 3      Ballard      0.06666667 0.0001121121  -2.639057  0.02895753
## 4 Beacon/Gtown/S.Park 0.08571429 0.0003731778  -2.367124  0.06076389
## HT.logit.prec  n  y
## 1      33.52878 278 39
## 2      32.25414 181 42
## 3      34.53333 555 37
## 4      16.45714 210 18
```

The smoothed estimates of  $p_i$  and  $\theta_i$  can be found in the smooth object returned by the function, and the direct estimates are stored in the HT object (without specifying survey weights, these are the simple binomial probabilities).

```
head(smoothed$smooth,n=1)
##      region      mean      var      median      lower      upper logit.mean
## 1 Auburn-North 0.1357317 0.000238308 0.1350914 0.1072711 0.1678374 -1.857599
##      logit.var logit.median logit.lower logit.upper
## 1 0.01726067 -1.856648 -2.118579 -1.603602
head(smoothed$HT,n=1)
##      region      HT.est      HT.var HT.logit.est HT.logit.var HT.logit.prec
## 1 Auburn-North 0.1402878 0.0004338385 -1.812902 0.02982513 33.52878
##      n y
## 1 278 39
```

We map the posterior mean estimates.

```
data(KingCounty)
toplot <- smoothed$smooth
mapPlot(data=toplot,geo=KingCounty,
variables=c("mean"),
labels=c("Posterior Mean"),by.data="region",by.geo="HRA2010v2_")
```

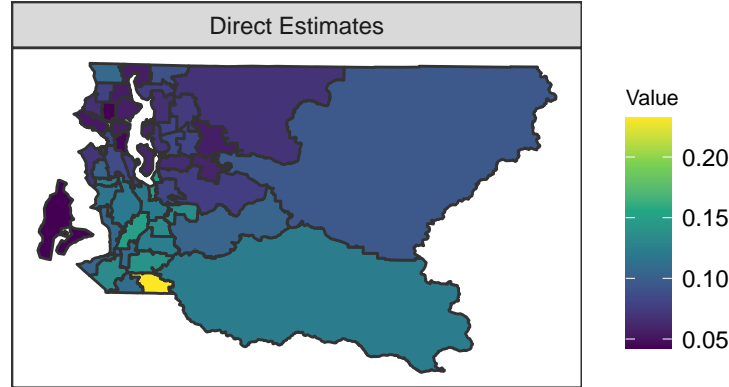


We map the Direct Estimates, which are available in the `smoothSurvey` fit.

```

topplot$HTest <- smoothed$HT$HT.est
mapPlot(data=topplot,geo=KingCounty,
variables=c("HTest"),
labels=c("Direct Estimates"),by.data="region",by.geo="HRA2010v2_")

```

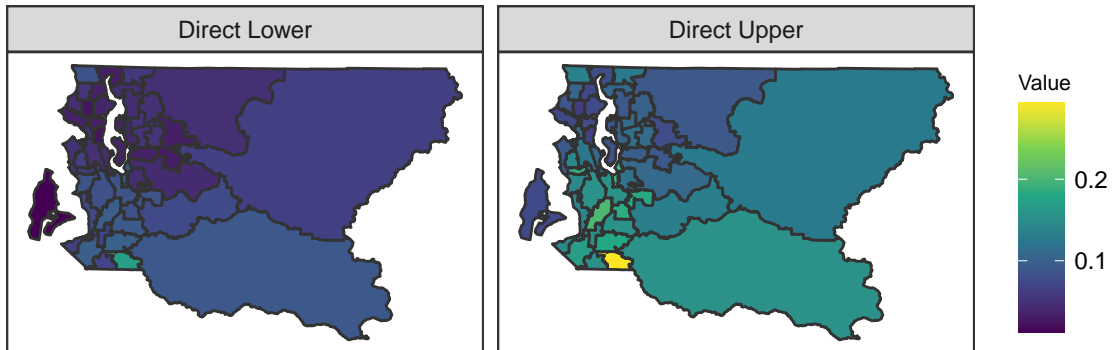


Now map the lower and upper endpoints of 95% CI for direct estimates

```

lo <- smoothed$HT$HT.est-1.96*sqrt(smoothed$HT$HT.var)
hi <- smoothed$HT$HT.est+1.96*sqrt(smoothed$HT$HT.var)
topplot$HTlower <- lo
topplot$HTupper <- hi
mapPlot(data=topplot,geo=KingCounty,
variables=c("HTlower","HTupper"),
labels=c("Direct Lower","Direct Upper"),by.data="region",by.geo="HRA2010v2_")

```



## Fit Fay-Herriot smoothing model, which acknowledges the design

We now acknowledge the design and fit the model

$$\hat{\theta}_i \sim N(\theta_i, \hat{V}_i)$$

with  $\hat{\theta}_i = \log[\hat{p}_i/(1 - \hat{p}_i)]$  where  $\hat{p}_i$  being the direct estimate and  $\hat{V}_i$  the variance of this estimate (where the design is acknowledged in the variance calculation) and

$$\theta_i = \log\left(\frac{p_i}{1 - p_i}\right) = \mu + \epsilon_i + \epsilon_i$$

with  $\epsilon_i \sim_{iid} N(0, \sigma^2)$ .

We put Amat=NULL to obtain an iid model only (i.e., the standard Fay-Herriot model without covariates).

```
FHmodel <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = NULL, responseType = "binary", responseVar = "hracode", clusterVar = "~1", CI = 0.95)
FHmodel$fit$summary.fixed[1:5]
##               mean          sd 0.025quant  0.5quant 0.975quant
## (Intercept) -2.666295 0.070691 -2.805911 -2.666151 -2.527495
FHmodel$fit$summary.hyper[1:5]
##               mean          sd 0.025quant  0.5quant 0.975quant
## Precision for region.struct 7.332147 2.554569  3.746102 6.882306 13.50861
sqrt(1/FHmodel$fit$summary.hyper[3:5])
##               0.025quant  0.5quant 0.975quant
## Precision for region.struct 0.5166664 0.3811826 0.2720788
```

Now extend the random effects structure to allow for BYM2 random effects.

```
svysmoothed <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = mat, responseType = "binary", responseVar = "hracode", clusterVar = "~1", CI = 0.95)
svysmoothed$fit$summary.fixed[1:5]
##               mean          sd 0.025quant  0.5quant 0.975quant
## (Intercept) -2.669854 0.04464686 -2.757531 -2.669871 -2.582078
svysmoothed$fit$summary.hyper[1:2,1:5]
##               mean          sd 0.025quant  0.5quant
## Precision for region.struct 11.4293385 4.2675762  5.3479916 10.6745247
## Phi for region.struct      0.8028236 0.1680026  0.3802848 0.8510944
##               0.975quant
## Precision for region.struct 21.8984850
## Phi for region.struct      0.9914791
sqrt(1/svysmoothed$fit$summary.hyper[1,3:5])
##               0.025quant  0.5quant 0.975quant
## Precision for region.struct 0.4324189 0.3060735 0.2136943
```

```
est <- data.frame(naive = smoothed$HT$HT.est,
weighted = svysmoothed$HT$HT.est,
smooth = smoothed$smooth$mean,
weightedsmooth = svysmoothed$smooth$mean)
var <- data.frame(naive = smoothed$HT$HT.var,
weighted = svysmoothed$HT$HT.var,
smooth = smoothed$smooth$var,
weightedsmooth = svysmoothed$smooth$var)
```

Setting Up

```
l1 <- range(est)
l2 <- range(var)
library(ggplot2)
g1 <- ggplot(est, aes(x = naive, y = smooth)) + geom_point() +
geom_abline(slope = 1, intercept = 0, color = "red") +
ggtitle("Naive Ests") + xlab("Direct") + ylab("Smoothed") + xlim(l1) + ylim(l1)
g2 <- ggplot(var, aes(x = naive, y = smooth)) + geom_point() +
geom_abline(slope = 1, intercept = 0, color = "red") +
```

```

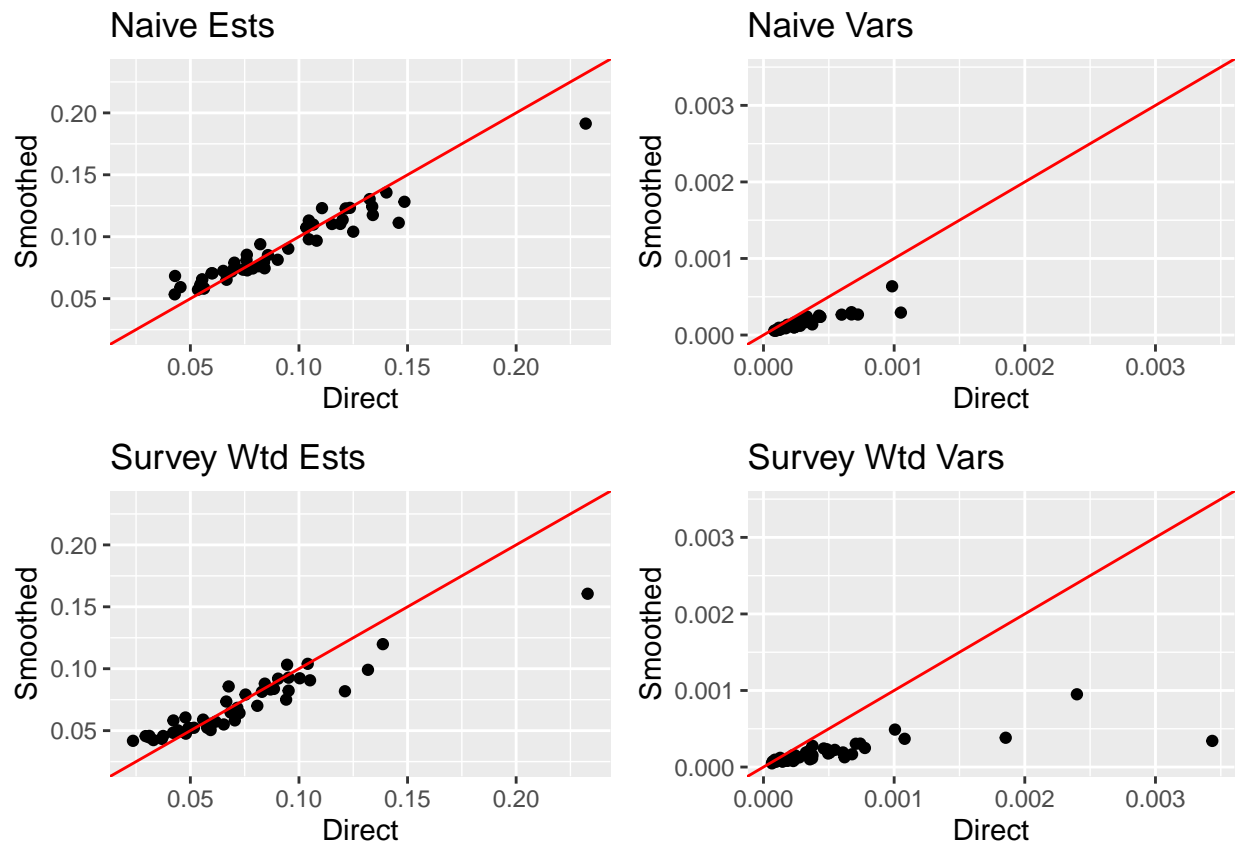
ggtitle("Naive Vars") + xlab("Direct") + ylab("Smoothed") + xlim(12) + ylim(12)
g3 <- ggplot(est, aes(x = weighted, y = weightedsmooth)) + geom_point() +
geom_abline(slope = 1, intercept = 0, color = "red") +
ggtitle("Survey Wtd Ests") + xlab("Direct") + ylab("Smoothed") + xlim(11) + ylim(11)
g4 <- ggplot(var, aes(x = weighted, y = weightedsmooth)) + geom_point() +
geom_abline(slope = 1, intercept = 0, color = "red") +
ggtitle("Survey Wtd Vars") + xlab("Direct") + ylab("Smoothed") + xlim(12) + ylim(12)
library(gridExtra)

```

```

grid.arrange(grobs = list(g1, g2, g3, g4), ncol = 2)

```



## SAE in Space and Time

When data consist of observations from different time periods, we can extend the framework to smooth estimates over both space and time. The space-time interaction terms are modeled by the type I-IV interactions – see Held (2000, Statistics in Medicine).

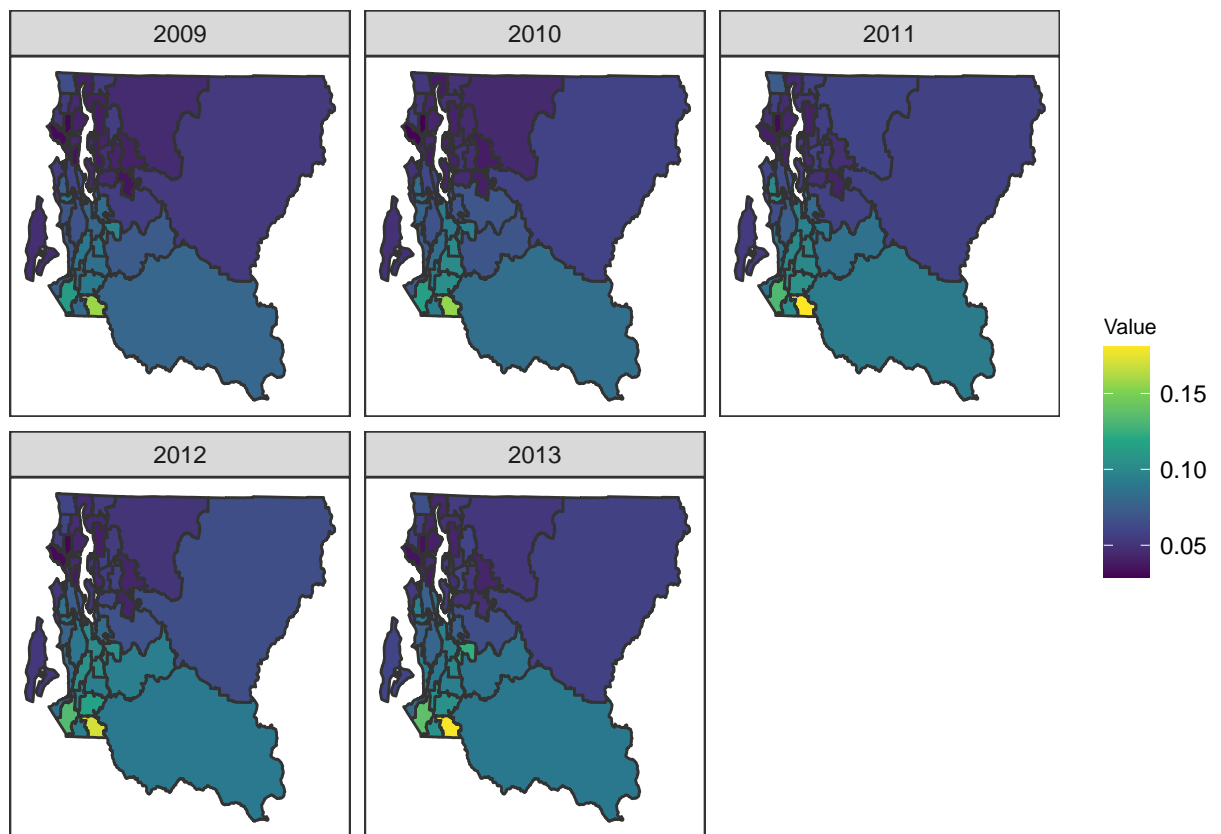
```

svysmoothed.year <- fitGeneric(data = BRFSS, geo = KingCounty, Amat = mat,
responseType = "binary", responseVar = "diab2", strataVar = "strata", weightVar = "rwt_llcp",
regionVar = "hrrcode", clusterVar = "~1", timeVar = "year", time.model = "rw1",
type.st = 1)

```

Maps of Posterior Means over Time

```
mapPlot(data = svysmoothed.year$smooth, geo = KingCounty, values = "mean",
variables = "time", by.data = "region", by.geo = "HRA2010v2_", is.long = TRUE)
```



## Final Comments

More materials can be found [here](#)

SUMMER has a Github page with the latest changes, see also:

[paper](#)