

# Small part of a large joint project with GNS Science, *ME* *Research*, and Resilient Organisations

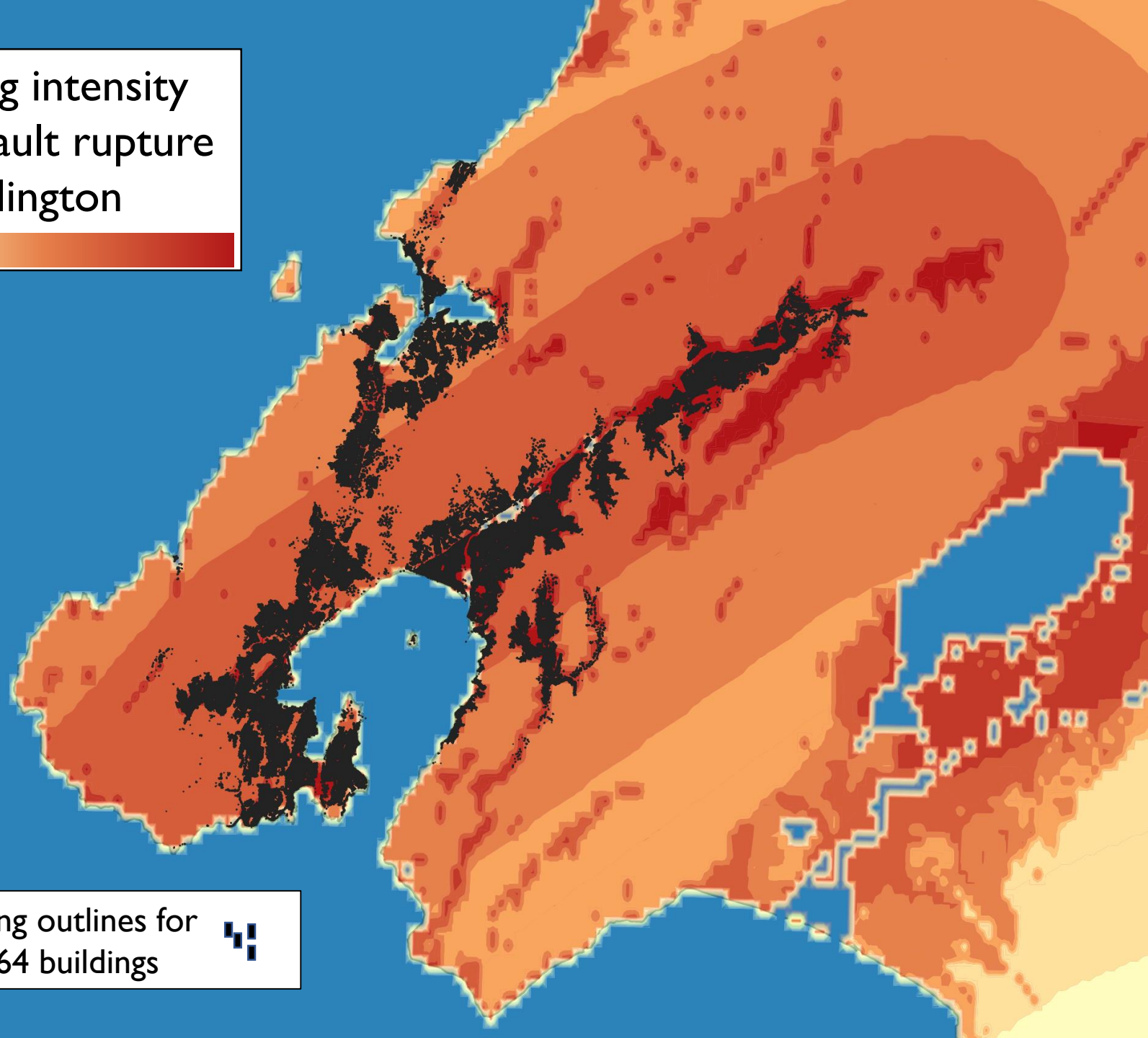
Data: 1000 'runs' of the *RiskScape* model which gives us 'Damage State' (0-4) for all 206,364 buildings

*NB: these results are from some dummy/test data.*

Shaking intensity  
for a fault rupture  
in Wellington



Building outlines for  
206,364 buildings





# Creating cordons around buildings

Building cordon estimate logic:

- Any building DS4 is cordoned off.
- Any of the cordoned off buildings that are over 3 storeys high, have a *height and unreinforced masonry*-dependent buffer around them.
- Any buildings in the buffer zone are also cordoned off.

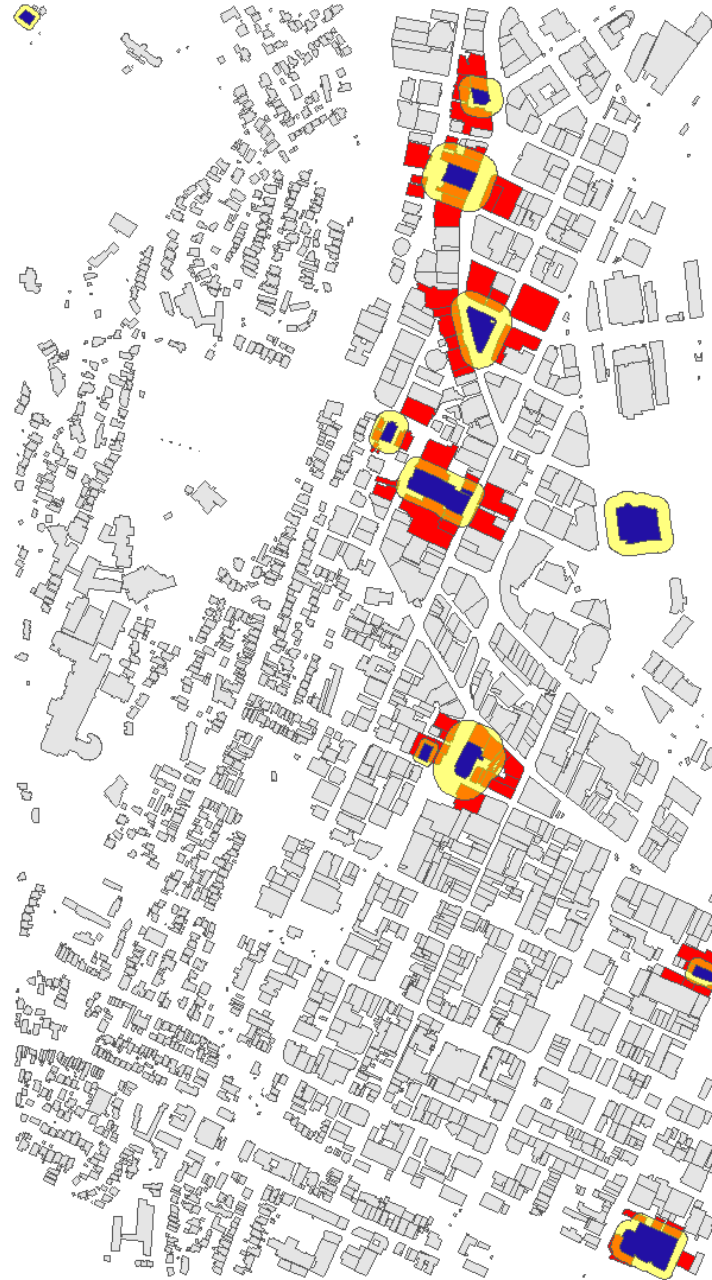
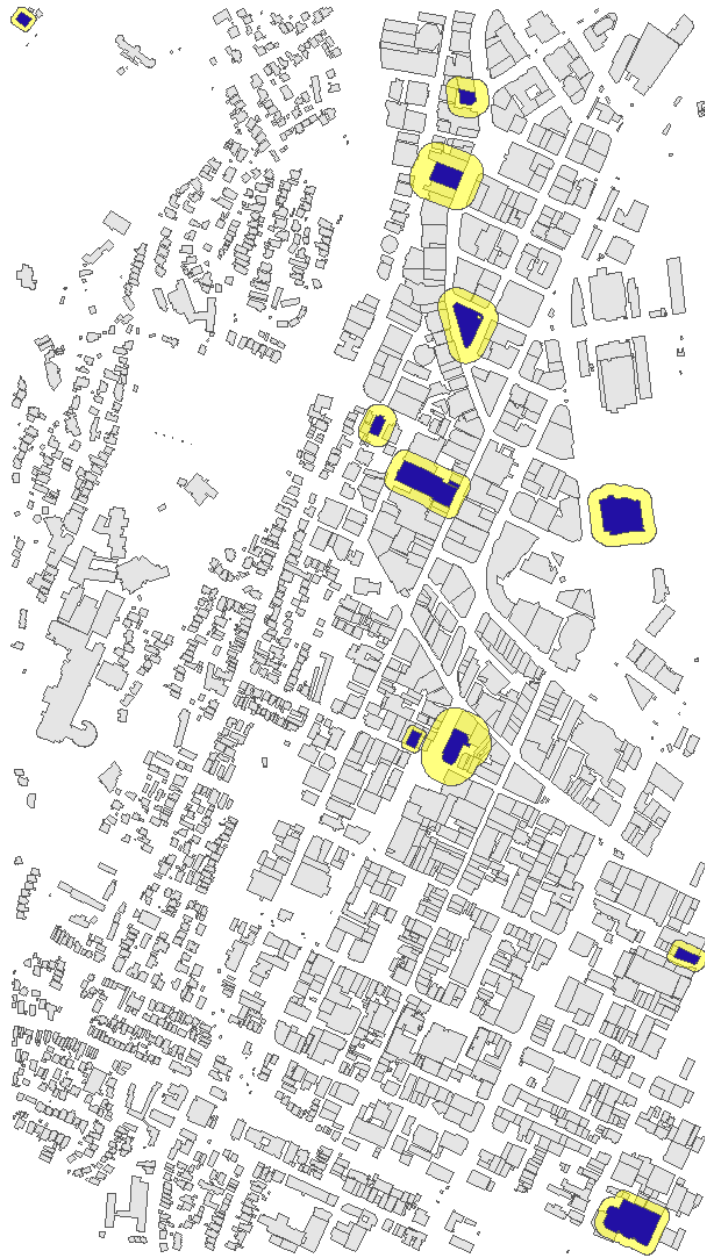
Original test data had only 10 runs (damage realisations).

Raw calculations took ~15mins in FME Workbench (ArcGIS visual workflow editor). But also required manual processing (results for each run that had to be combined) and each run needed its own damage state data input shapefile.

# Rough R code

```
library(tidyverse)
library(sf)
# load building spatial file – with buffer widths and unreinforced masonry info
buildings <- st_read("buildings.shp")
damagestates <- readRDS("damagestates_1000.rds")
# single run example – though in reality I wrote a function that looped through
rundata <- buildings %>%
  left_join(damagestates) %>%
  select(BuildingID, Run_1) %>%
  rename(DamageState = Run_1)
# find which buildings need buffer zones and create buffers as a single spatial object
buffers <- rundata %>%
  filter(DamageState > 3 & Storeys >= 3) %>%
  st_buffer(dist=.$BufferWidth) %>%
  st_union() %>% st_cast()
# find which buildings intersect the buffer zones
results <- buildings %>% select(BuildingID)
results$inside_cordon <- st_intersects(buildings, buffers, sparse = FALSE)
```





Of the 206,364 buildings  
→ ~20 with buffer zones  
→ ~100 cordoned off

Time for one run/damage  
realisation

R: ~6 s

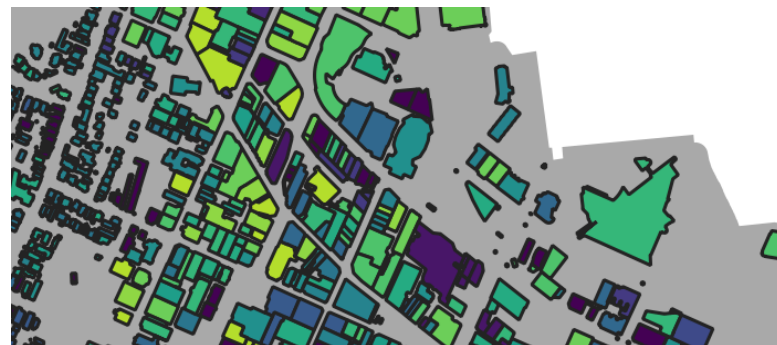
FME workbench: ~2min  
(not including manual  
processing)

*NB: these results are from some  
dummy/test data.*

# Plotting with sf in R (so easy!)

Colouring by width of buffer

```
ggplot()+  
  geom_sf(data=basemap, fill = grey) +  
  geom_sf(data=buildings, aes(fill=Bufferwidth)) +  
  scale_fill_gradientn(colors = viridis::viridis(20))
```



Stacking a few things

```
ggplot()+  
  geom_sf(data=buildings) +  
  geom_sf(data=results %>% filter(inside_cordon==TRUE),  
          fill = "red") +  
  geom_sf(data=buffers, fill= "yellow", alpha=0.5) +  
  geom_sf(data=rundata %>% filter(DamageState > 3 & Storeys >= 3),  
          fill = "#2210A4")
```

