

Report

Signal detection of spontaneous medical device reports over time accounting for multiple comparisons

Ty Stanford et al.

Table of contents

1	Set up	2
1.1	Packages	2
1.2	Load data	2
2	Methods	4
2.1	Data aquisition	4
3	The signal detection statistics over time	5
3.1	Data	5
3.2	Reporting odds ratio (ROR)	6
3.3	BCPNN IC	6
3.4	maxSPRT	6
4	Analysis choices:	7
5	Plots	8
6	Session information	18

1 Set up

1.1 Packages

```
suppressPackageStartupMessages({  
  library("readr")  
  library("dplyr")  
  library("tidyr")  
  library("forcats")  
  library("lubridate") # way to handle dates better than default R way  
  library("ggplot2")  
  library("ggthemes")  
  library("ggrepel")  
  library("knitr")  
  library("gsDesign")  
  library("arrow")  
})
```

Warning: package 'dplyr' was built under R version 4.2.3

```
col_pal <- c("darkorange", "cyan4", "purple")
```

1.2 Load data

```
sra_cum_bcpnn <- read_parquet("out/sra_cum_bcpnn.parquet")  
sra_cum_bcpnn_mc_adj <- read_parquet("out/sra_cum_bcpnn_mc_adj.parquet")  
sra_cum_maxsprt <- read_parquet("out/sra_cum_maxsprt.parquet")  
  
sra <-  
  bind_rows(  
    sra_cum_bcpnn %>% mutate(stat = "BCPNN"),  
    sra_cum_bcpnn_mc_adj %>% mutate(stat = "BCPNN (MCadj)"),  
    sra_cum_maxsprt %>% mutate(stat = "maxSPRT")  
  ) %>%  
  select(stat, everything())  
  
sra <-
```

```
sra %>%  
mutate(  
  test_stat = if_else(stat == "maxSPRT", maxllr, ci_lo),  
  test_thresh = if_else(stat == "maxSPRT", cv, 0),  
  rr_stat = if_else(stat == "maxSPRT", rre, 2 ^ est)  
)
```

```
thresholds <- sort(unique(sra[["thresh"]]))  
length(thresholds)
```

```
[1] 19
```

2 Methods

2.1 Data acquisition

The data is thanks to [curtis-murray](#) at his [MedicalDevicesNLP](#) repo

- Natural language processing of the TGA spontaneous reports of medical device database (DAEN)
- Each record has an estimate of $P(\text{topic} == \text{"pain"} \mid \text{Level}, \text{Doc})$ using hierarchical stochastic block modelling (hSBM)
- $P(\text{topic} == \text{"pain"} \mid \text{Level}, \text{Doc})$ estimates for each record are roughly interpreted as the proportion of the NLP analysed free text that is considered as using/describing words related to pain

And example record and processing values:

- [to include here]

3 The signal detection statistics over time

We will consider the three signal detection statistics below:

- Reporting Odds Ratio (ROR),
- Bayesian Confidence Propagation Neural Network Information Component (BCPNN IC with MCMC CIs), and
- the maxSPRT statistic

However, first let's outline the data available.

3.1 Data

Signal detection of disproportionate adverse events (AEs) will often have tabulated count data accumulated over time. The data at time point t can be summarised as below:

	AE(s) X	AE(s) \bar{X}
Target exposure	a_t	b_t
Comparator exposure	c_t	d_t

where

- AE(s) X is the set of AEs (or singular AE) of interest,
- AE(s) \bar{X} is the complementary set to the AEs of interest,
- *Target exposure* is the medical device(s) of interest,
- *Comparator exposure* is the medical devices to which the *Target exposure* is being compared, and
- a_t , b_t , c_t and d_t (all $\in \mathbb{Z}^+$) are the respective counts of AEs recorded up until (i.e., cumulative) time t .

In the motivating example of the pelvic mesh device, the contingency table can be written more specifically as

	AEs pain	AEs not pain
Pelvic mesh	a_t	b_t
Comparator exposure	c_t	d_t

where

- *AEs pain* is the count of AEs that contain “pain” themes greater or equal to some pre-specified threshold $p_t \in (0,1)$ as estimated by the hSBM (that is, $P(\text{topic} == \text{"pain"} | \text{Level}, \text{Doc}) \geq p_t$), and
- *Comparator exposure* can be any relevant set of medical devices to compare the pelvic mesh to (e.g., hernia mesh or all other mesh devices or all other devices).

3.2 Reporting odds ratio (ROR)

ROR with $100(1 - \alpha/2)\%$ confidence intervals

$$\hat{\text{ROR}} = \frac{\frac{a_t}{a_t + b_t}}{\frac{c_t}{c_t + d_t}}$$

3.3 BCPNN IC

BCPNN IC using the *maximum a posteriori* (m.a.p.) central estimate of the IC with MCMC simulation of the exact empirical distribution for $100(1 - \alpha/2)\%$ confidence (credible) regions of [Noren \(2006\)](#)

3.4 maxSPRT

The maximised log-likelihood ratio statistic

4 Analysis choices:

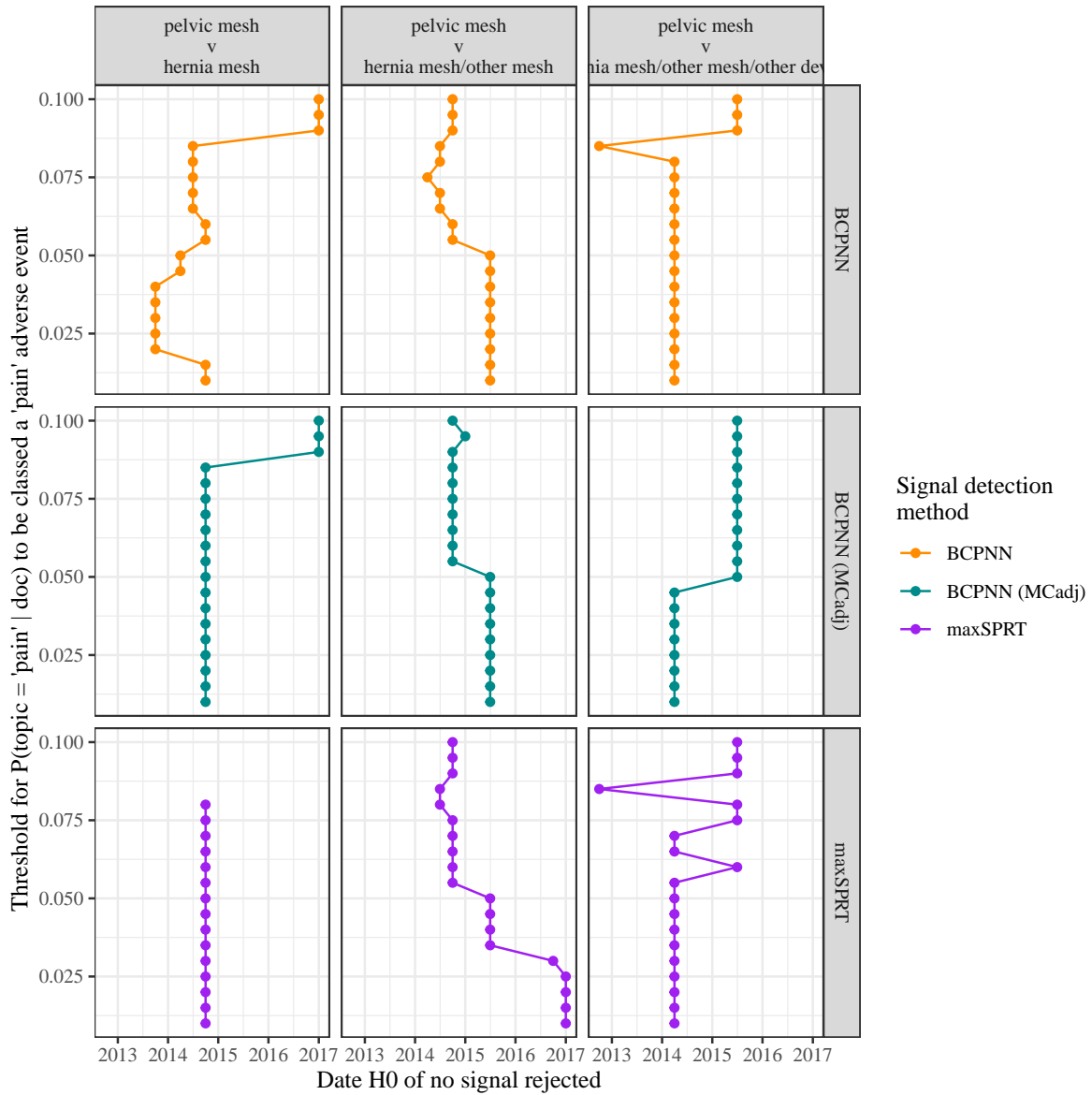
- Data structures - cumulative vs snapshot
- Threshold choose
- How many “looks”
- how to choose alpha spending
- sample size limitations for maxspt - not an issue now can use MCMC method of EmpiricalCalibration

5 Plots

```
date_signif_dat <-
  sra %>%
  group_by(stat, grps, dat_type, thresh) %>%
  arrange(dte) %>%
  dplyr::filter(reach_sig) %>%
  dplyr::filter(row_number() == 1) %>%
  ungroup() %>%
  arrange(stat, grps, dat_type, thresh)

signif_plt <-
  date_signif_dat %>%
  ### only keep pelvic mesh as target vs whatever comparator
  dplyr::filter(grepl("^.*pelvic.* v ", grps)) %>%
  mutate(
    grps = gsub(" v ", "\nv\n", grps),
    grps = gsub("\\([a-z]\\)", "", grps),
    grps = gsub("_", " ", grps),
    grps = fct_inorder(grps)
  )

signif_plt %>%
  arrange(grps, thresh) %>%
  ggplot(., aes(x = dte_reach_sig, y = as.numeric(thresh), col = stat)) +
  geom_point() +
  geom_path(aes(group = stat)) +
  scale_colour_manual(values = col_pal) +
  # scale_colour_tableau(palette = "Color Blind", direction = -1) +
  facet_grid(stat ~ grps) +
  # facet_wrap( ~ grps, ncol = 1) +
  theme_bw() +
  theme(text = element_text(family = "serif")) +
  labs(
    x = "Date H0 of no signal rejected",
    y = "Threshold for P(topic = 'pain' | doc) to be classed a 'pain' adverse event",
    col = "Signal detection\nmethod"
  )
```

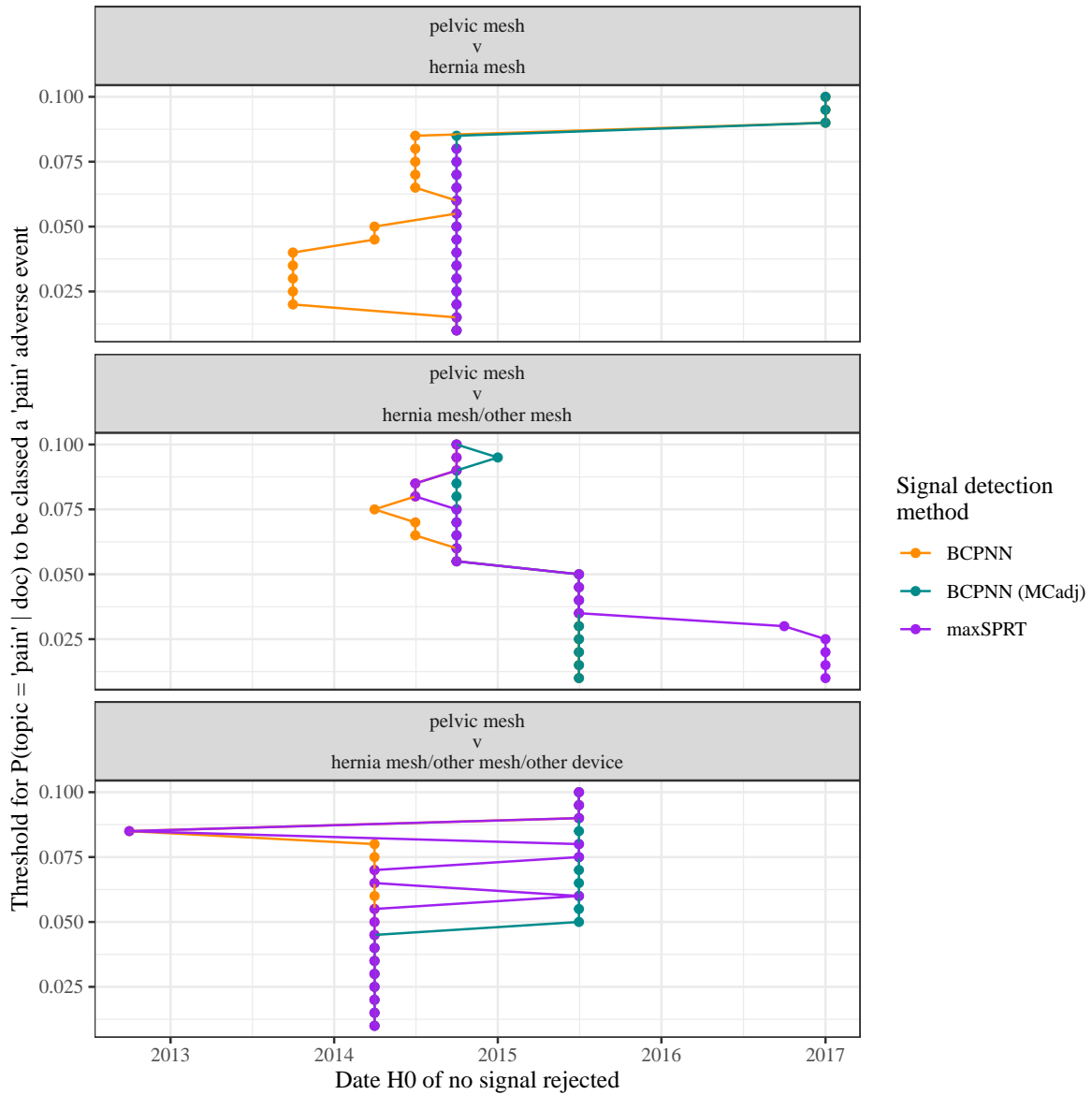
```
ggsave(
  filename = "fig/time_to_signal_method_facets.png",
  dpi = 900, width = 9, height = 9
)

signif_plt %>%
  arrange(grps, thresh) %>%
```

```

ggplot(., aes(x = dte_reach_sig, y = as.numeric(thresh), col = stat)) +
geom_point() +
geom_path(aes(group = stat)) +
scale_colour_manual(values = col_pal) +
# scale_colour_tableau(palette = "Color Blind", direction = -1) +
# facet_grid(stat ~ grps) +
facet_wrap( ~ grps, ncol = 1) +
theme_bw() +
theme(text = element_text(family = "serif")) +
labs(
  x = "Date H0 of no signal rejected",
  y = "Threshold for P(topic = 'pain' | doc) to be classed a 'pain' adverse event",
  col = "Signal detection\nmethod"
)

```



```

ggsave(
  filename = "fig/time_to_signal_method_overlay.png",
  dpi = 900, width = 5, height = 9
)

```

```

sra_stat_plt <-
  sra %>%
    # keep only subset of thresholds (too many colours otherwise)
    dplyr::filter(thresh %in% sprintf("%.3f", seq(0.02, 0.08, by = 0.02))) %>%
    ### only keep pelvic mesh as target vs whatever comparator
    # dplyr::filter(grepl("^.*pelvic.* v ", grps)) %>%
    dplyr::filter(grepl("^\\(a\\)", grps)) %>%
    mutate(
      grps = gsub(" v ", "\\nv\\n", grps),
      grps = gsub("\\([a-z]\\)", "", grps),
      grps = gsub("_", " ", grps),
      grps = fct_inorder(grps)
    )

thresholds <- sort(unique(sra_stat_plt[["thresh"]]))
length(thresholds)

```

[1] 4

```

thresh_scale <- rev(hcl.colors(length(thresholds), "SunsetDark"))
# thresh_scale <- rev(hcl.colors(length(thresholds) + 1, "Inferno"))[-1]

sra_stat_plt %>%
  mutate(
    reach_sig_alpha = ifelse(reach_sig, 1, 0.8),
    `P(topic = 'pain') threshold` = thresh,
    `Test` = if_else(stat == "maxSPRT", "Maximised LLR > CV", "IC lower 95% > 0")
  ) %>%
  ggplot(
    .,
    aes(
      x = dte,
      y = test_stat,
      col = stat,
      group = interaction(stat, thresh)
      # alpha = reach_sig_alpha
    )
  ) %>%
  geom_hline(aes(yintercept = test_thresh), col = "black") %+% # null value
  geom_vline(aes(xintercept = dte_reach_sig, col = stat), alpha = 0.5) %+% # sig first rea

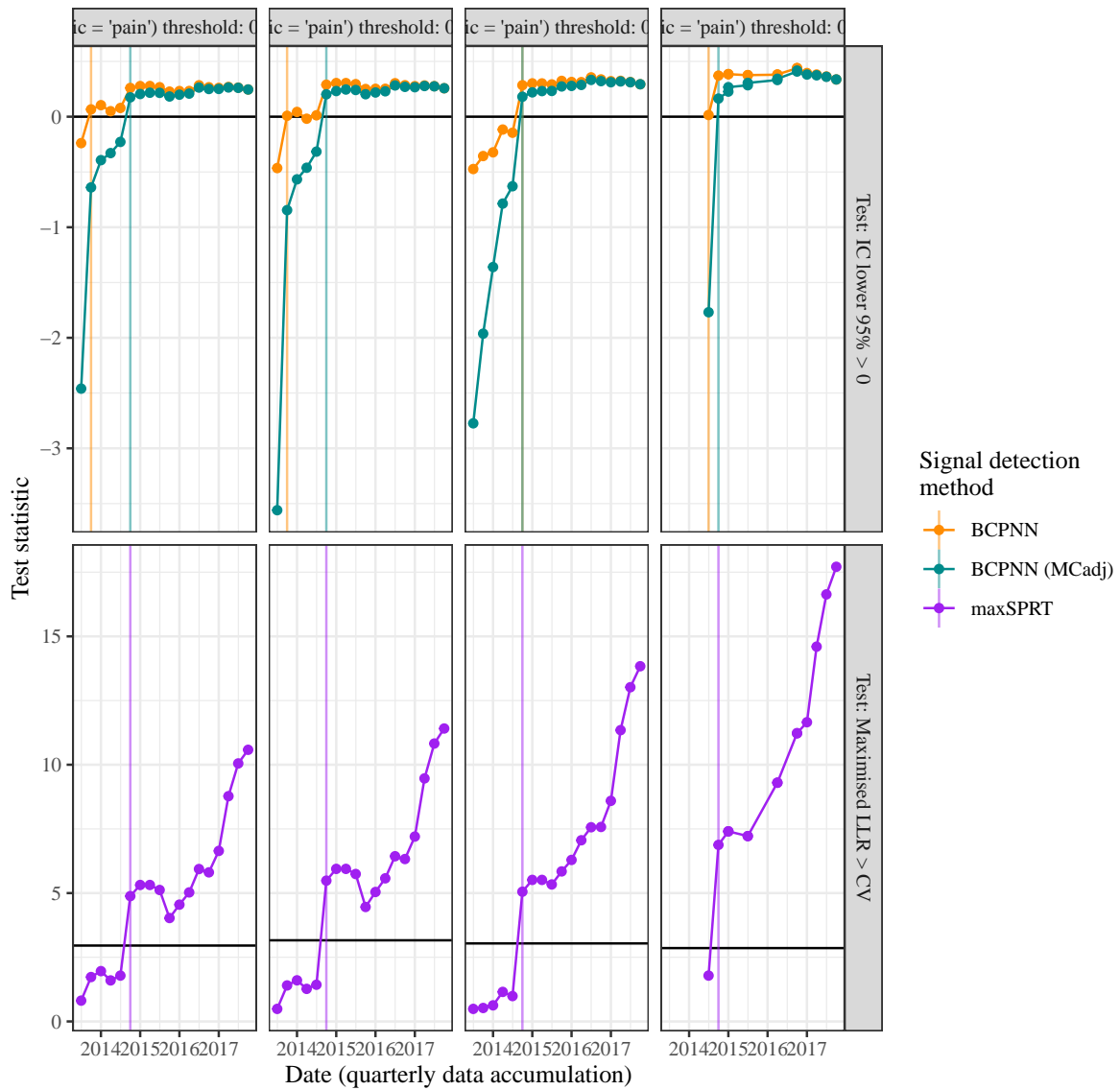
```

```

geom_line() %+%
geom_point() %+%
# geom_ribbon(alpha = 0.05, lty = 2) %+%
# facet_wrap(~ grps, scales = "free_y", ncol = 1) %+%
facet_grid(
  `Test` ~ `P(topic = 'pain') threshold`,
  scales = "free_y",
  labeller = label_both
) %+%
labs(
  subtitle = "Pelvic mesh v hernia mesh",
  y = "Test statistic",
  x = "Date (quarterly data accumulation)",
  col = "Signal detection\nmethod"
) %+%
# scale_y_continuous(limits = c(NA, 6)) %+%
scale_colour_manual(values = col_pal) %+%
# scale_colour_tableau(palette = "Color Blind", direction = -1) +
theme_bw() %+%
theme(text = element_text(family = "serif"))

```

Pelvic mesh v hernia mesh



```

ggsave(
  filename = "fig/pelvic_v_hernia_sig_detect_over_time.png",
  dpi = 900, width = 10, height = 8
)

```

```

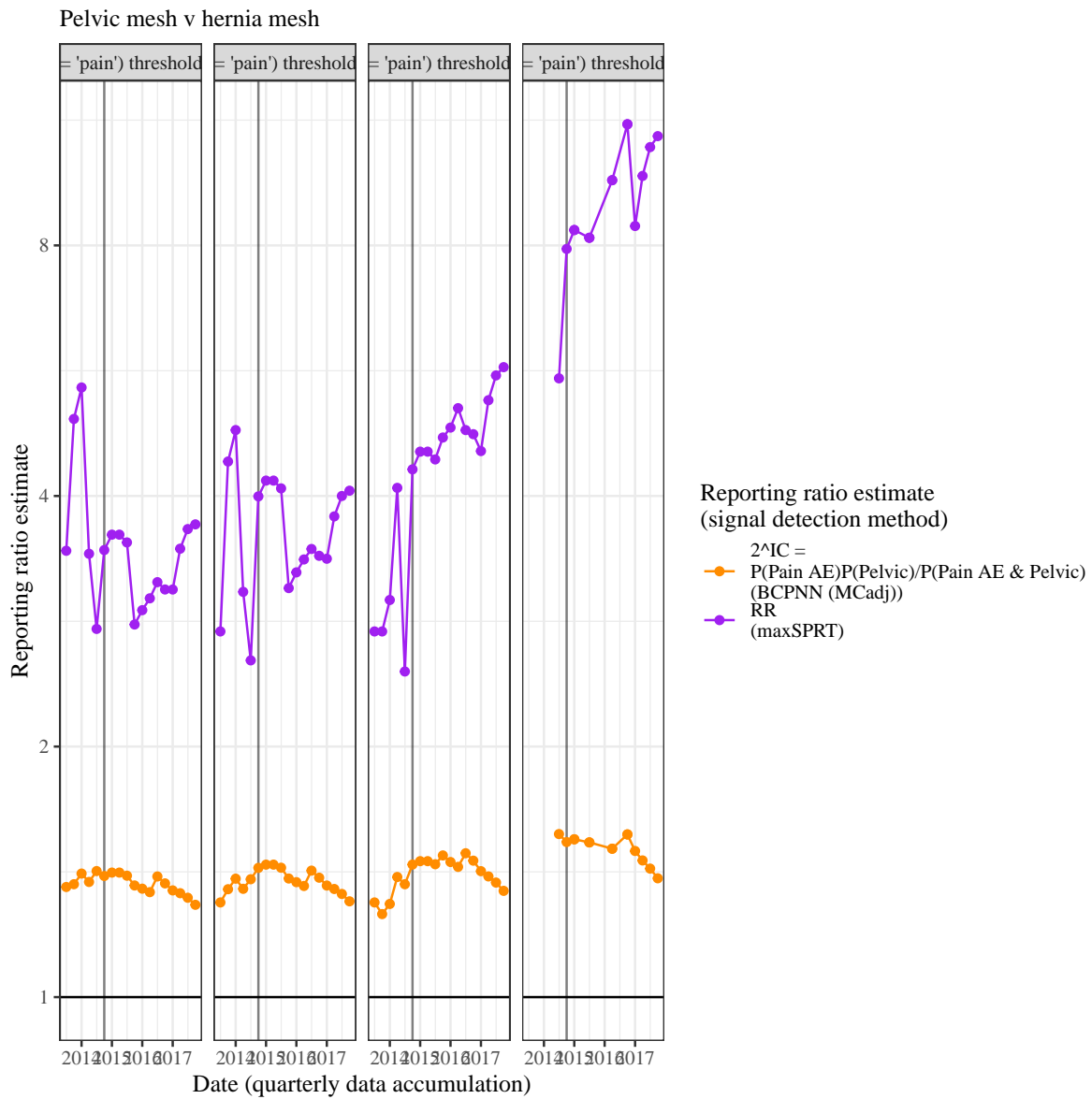
sra_stat_plt %>%
  dplyr::filter(stat != "BCPNN") %>%
  mutate(
    stat2 =
      paste0(
        ifelse(stat == "maxSPRT", "RR", "2^IC =\nP(Pain AE)P(Pelvic)/P(Pain AE & Pelvic)")
        "\n(", stat, ")")
      ),
    reach_sig_alpha = ifelse(reach_sig, 1, 0.8),
    `P(topic = 'pain') threshold` = thresh,
    `Test` = if_else(stat == "maxSPRT", "Maximised LLR > CV", "IC lower 95% > 0"),
    `Statistic calculation method` = stat
  ) %>%
  ggplot(
    .,
    aes(
      x = dte,
      y = rr_stat,
      col = stat2,
      group = stat2
      # group = interaction(stat2, thresh)
      # alpha = reach_sig_alpha
      # ymin = ci_lo,
      # ymax = ci_hi
    )
  ) %+%
  geom_hline(aes(yintercept = 1), col = "black") %+% # null value
  geom_vline(aes(xintercept = dte_reach_sig), alpha = 0.5) %+% # sig first reached
  geom_line() %+%
  geom_point() %+%
  # geom_ribbon(alpha = 0.05) %+%
  facet_wrap(~ `P(topic = 'pain') threshold`, nrow = 1, labeller = label_both) %+%
  # facet_grid(
  #   `Statistic calculation method` ~ `P(topic = 'pain') threshold`,
  #   labeller = label_both
  # ) %+%
  labs(
    subtitle = "Pelvic mesh v hernia mesh",
    y = "Reporting ratio estimate",
    x = "Date (quarterly data accumulation)",
    col = "Reporting ratio estimate\n(signal detection method)"
  )

```

```

) %+%
scale_y_continuous(trans = "log2") %+%
scale_colour_manual(values = col_pal[-2]) %+%
theme_bw() %+%
theme(text = element_text(family = "serif"))

```




```
ggsave(  
  filename = "fig/pelvic_v_hernia_rr_est_over_time.png",  
  dpi = 900, width = 12, height = 5  
)
```

6 Session information

```
# Sys.info()[!(names(Sys.info()) %in% c("login", "nodename"))] %>%  
#   as.data.frame(.)  
format(Sys.time(), '%d %b %Y')
```

```
[1] "10 Aug 2023"
```

```
sessionInfo()
```

```
R version 4.2.2 (2022-10-31 ucrt)  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
Running under: Windows 10 x64 (build 19045)
```

```
Matrix products: default
```

```
locale:
```

```
[1] LC_COLLATE=English_Australia.utf8  LC_CTYPE=English_Australia.utf8  
[3] LC_MONETARY=English_Australia.utf8 LC_NUMERIC=C  
[5] LC_TIME=English_Australia.utf8
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] arrow_11.0.0.2  gsDesign_3.4.0  knitr_1.42      ggrepel_0.9.3  
[5] ggthemes_4.2.4  ggplot2_3.4.1   lubridate_1.9.2 forcats_1.0.0  
[9] tidyr_1.3.0     dplyr_1.1.2     readr_2.1.4
```

```
loaded via a namespace (and not attached):
```

```
[1] Rcpp_1.0.10      pillar_1.9.0    compiler_4.2.2  tools_4.2.2  
[5] bit_4.0.5        digest_0.6.31   jsonlite_1.8.4  evaluate_0.20  
[9] lifecycle_1.0.3  tibble_3.2.1    gtable_0.3.1    timechange_0.2.0  
[13] pkgconfig_2.0.3  rlang_1.1.1     cli_3.6.0       rstudioapi_0.14  
[17] yaml_2.3.7       xfun_0.37       fastmap_1.1.0   stringr_1.5.0  
[21] withr_2.5.0      systemfonts_1.0.4 generics_0.1.3  vctrs_0.6.3  
[25] hms_1.1.2        bit64_4.0.5     grid_4.2.2      tidyselect_1.2.0  
[29] glue_1.6.2       R6_2.5.1        textshaping_0.3.6 fansi_1.0.4  
[33] rmarkdown_2.20   farver_2.1.1    tzdb_0.3.0      purrr_1.0.1
```

[37]	magrittr_2.0.3	scales_1.2.1	ellipsis_0.3.2	htmltools_0.5.4
[41]	assertthat_0.2.1	xtable_1.8-4	colorspace_2.1-0	ragg_1.2.5
[45]	labeling_0.4.2	utf8_1.2.3	stringi_1.7.12	munsell_0.5.0