Visualisations of the **bandgaps** dataset

Taha Ahmed

June 30, 2023

Here's my outline for a plotting function for the band gap plots:

1. User calls the function and specifies i) which materials, ii) at which pH (for all of them), and iii) which secondary scale to use (here we should leverage our `refelectrodes` package).

2. Internally, the function should recalculate the CB and VB levels based on the supplied pH value, assuming the semiconductors follow Nernstian behaviour. I am not sure how to best handle non-Nernstian semiconductors in that case.

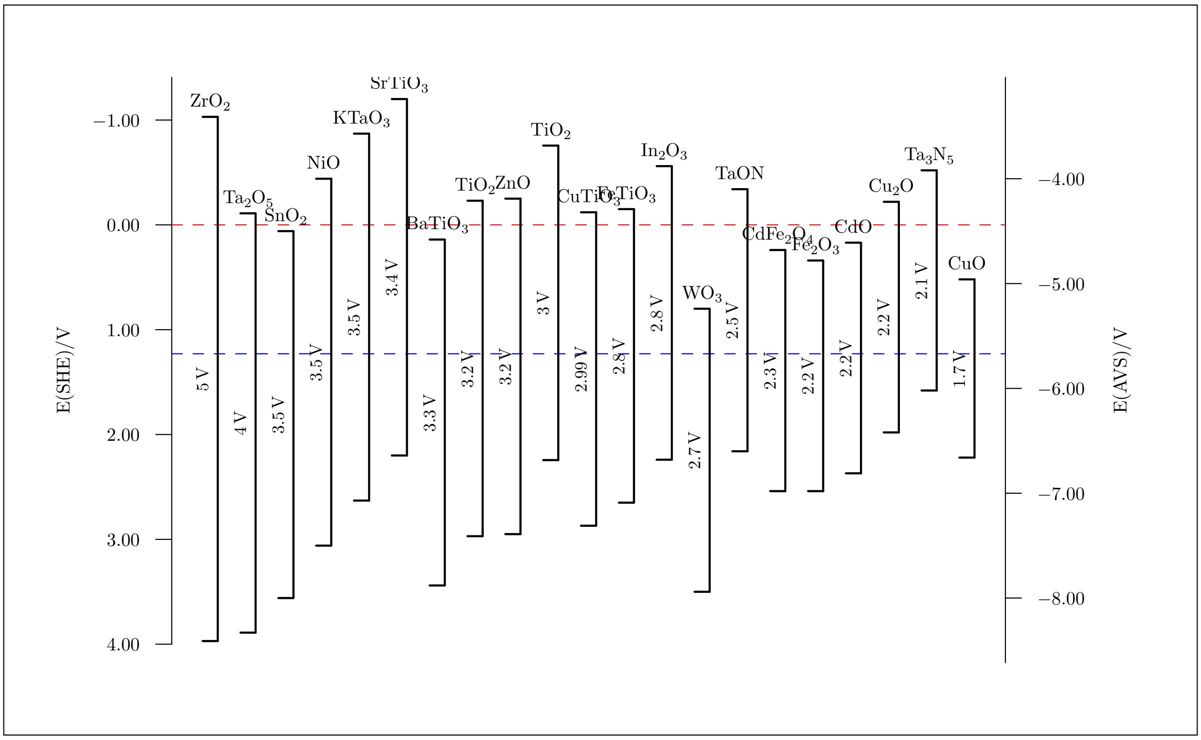
3. Finally, the plot needs to figure out the correct scaling and direction of the secondary scale. This turns out to be pretty easy with `sec.axis` in ggplot2, but slightly more cumbersome with the base plot package.

Not all of these points are implemented yet. In particular, a function that recalculates CB/VB levels based on pH assuming Nernstian behaviour has not been implemented yet. But other than that, the points above or mostly implemented in the base plot or ggplot2 plotting functions, which we will explore a little bit more below.

Band gaps plotted with the base plot package

For the R base plot package, this package implements the function `plot_bandgaps()`. The plot below can then be created with a single line of code, like this:

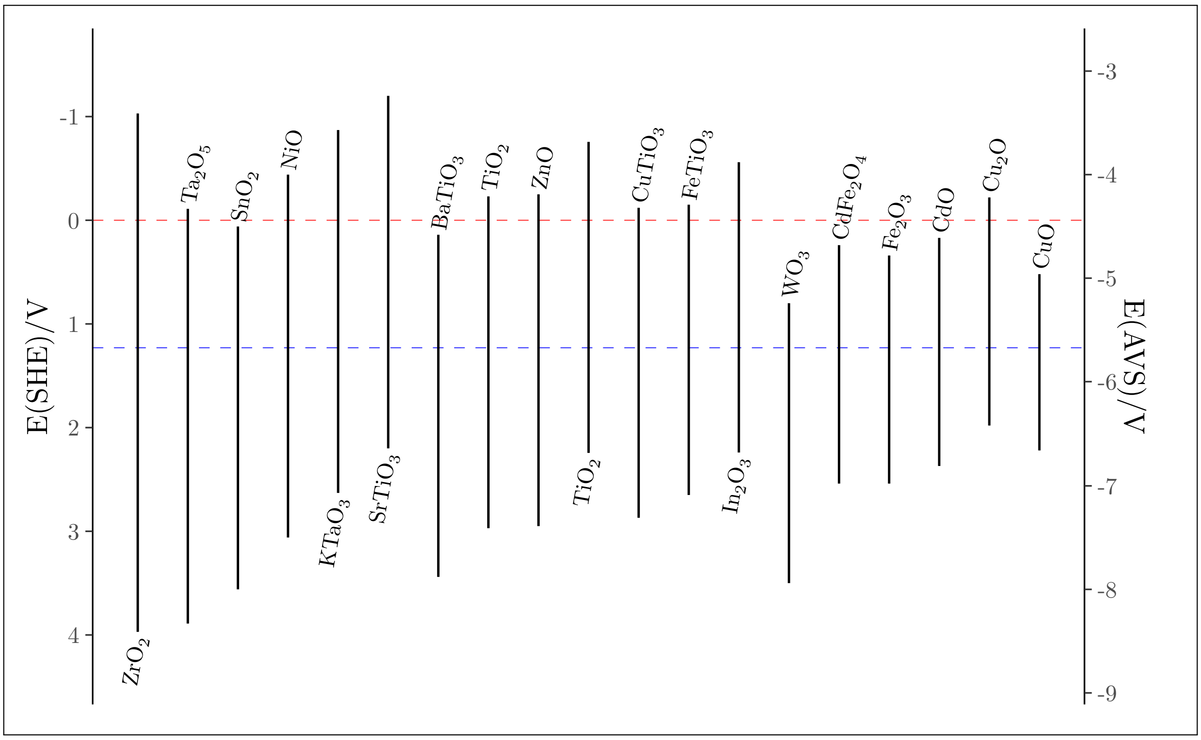
```r
plot_bandgaps(bandgaps::semiconductors %>% filter(Nernstian == TRUE))
```



This works alright, with the plot size adapting fairly well to the number of included semiconductors, or to the absence of the secondary y-axis. The plot lacks a good way to handle placement and overlap of the labels, though.

Band gaps plotted with ggplot2

For expediency, since this is the first version of this vignette, I decided to represent the band gap with just a vertical line instead of the traditional horseshoe. (Could we define that as an object? Thus only needing to specify one edge and its length to draw it? Essentially creating a ggplot2 geom?)
With a rather small effort we managed to have the labels for the highest CB edges below the VB instead of above the CB (to avoid overcrowding the top part of the plot). I also experimented with `ggrepel` for the labels, but found that using `geom_text_repel()` made the labels repel each other but not the line intervals themselves.
So here's the band gaps dataset plotted using ggplot2:



And this code chunk was used to generate it (this should be functionalised!):

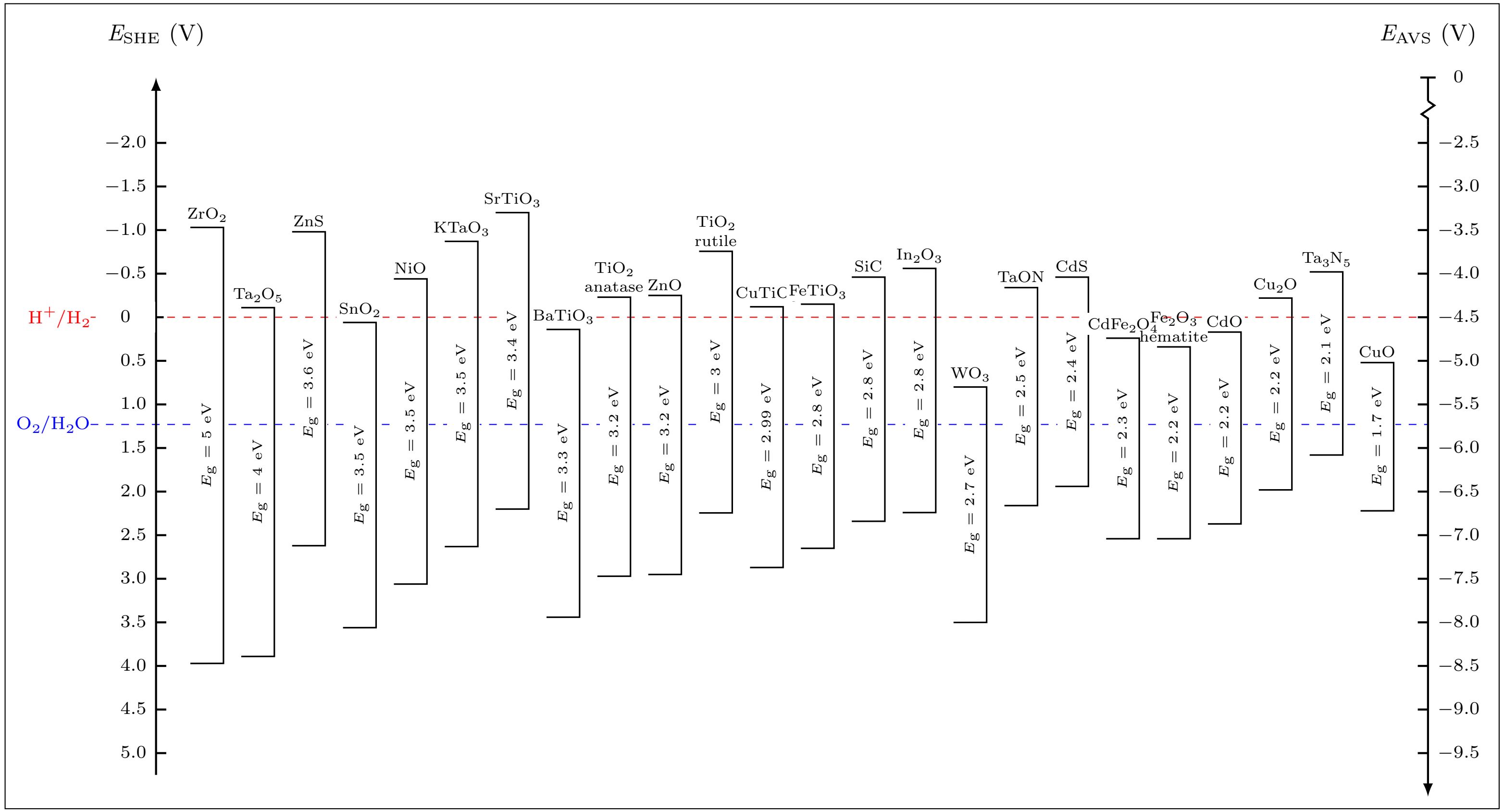
```r
ggplot_materials <-
   bandgaps::semiconductors %>%
   # for recalculation of pH, only display Nernstian semiconductors
   filter(class == "oxide")
ggplot_materials <-
   ggplot_materials %>%
   # add column with x variable
   # NOTE: do this as the last step to avoid weird visual gaps
   # 0.4 is the x-value of the first line, stepsize 1.0
   mutate(x = seq(0.4, dim(ggplot_materials)[1], by = 1.0))
ggplot(data = ggplot_materials) +
   # water redox lines
   geom_hline(yintercept = 0.00, colour = "red", linetype = "dashed", size = 0.3) +
   geom_hline(yintercept = 1.23, colour = "blue", linetype = "dashed", size = 0.3) +
   # band gaps visualised as vertical lines
   geom_linerange(
      aes(x = x, ymin = CB, ymax = VB),
      size = 0.8) +
   # label each compound
   geom_text(
      aes(
         x = x,
         label = paste0("\\ch{", formula, "}"),
         # can we put the label at VB if CB > than some value?
         # also add a constant space between CB/VB and label
         y = ifelse(CB > -0.5, CB - 0.05, VB + 0.05),
         # left-adjust labels at CB, and right-adjust labels at VB
         hjust = ifelse(CB > -0.5, 0, 1)),
      size = 3,
      vjust = 0.5,
      angle = 80) +
   # flip primary y-axis and create secondary AVS axis
   scale_y_continuous(
      name = "E(SHE)/V",
      # make sure each unit step is marked
      breaks = seq(-4, 6),
      # expand() parameters are
      # a multiplicative and an absolute expansion
      # we add some space to make sure the labels of the largest Eg fits
      expand = c(0, 0.65),
      trans = "reverse",
      sec.axis =
         # note the initial minus sign
         # which makes the AVS scale go positive up
         sec_axis(
            ~ -. + refelectrodes::as.SHE(0, scale = "AVS"),
            name = "E(AVS)/V")) +
   # a more suitable theme
   theme_classic() +
   # completely remove the x-axis
   theme(
      axis.text.x = element_blank(),
      axis.title.x = element_blank(),
      axis.line.x = element_blank(),
      axis.ticks.x = element_blank())
```

Band gaps plotted with TikZ

This is the original version of this plot and the dataset. In fact, this dataset started out as electrochemical potentials hard-coded in the TikZ code. But despite that lack of adaptability, the visual quality of the plot surpasses both the base plot and the ggplot2 above (perhaps not for much longer). In fact, a slightly modified version of this plot was published in Zhu [1] and Zhu [2, p. 397].

There is a lot of LaTeX and TikZ code behind this plot (have a look at the source file for this vignette if you like). Nonetheless, I have managed to make the code slightly more useful by integrating the dataset into the TikZ code using a fairly compact chunk of R code.

Still, the TikZ code is not really dynamic. For example, the primary and secondary y-axis scales are hard-coded in the TikZ code. For future development, I think it is more fruitful to work on the ggplot2 approach above rather than this TikZ approach.



References

[1] Jiefang Zhu. “Photo-catalytic Hydrogen Production”. In: *Encyclopedia of Sustainability Science and Technology.* Ed. by Robert A. Meyers. New York: Springer-Verlag, 2012, pp. 7881–7901.

[2] Jiefang Zhu. “Photocatalysts for Hydrogen Production”. In: *Advanced Materials For Clean Energy.* Ed. by Qiang Xu and Tetsuhiko Kobayashi. Taylor & Francis Group, 2015, pp. 391–419.

Source: `diagram.Rnw`
git refs: `(HEAD -> master)`
git hash: `11246a70f2abe6d2d671a5326d167ce37b6cfbb8`
git author: taha@luxor
author email: taha@chepec.se
commit date: 2018-03-12 14:21:31 +0100
compile date: 2023-06-30 00:04:04

- R version 4.1.3 (2022-03-10), `x86_64-pc-linux-gnu`
- Running under: `Ubuntu 22.04.2 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0`
- LAPACK: `/usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: bandgaps 0.1.1.9000, common 0.0.2, dplyr 1.0.10, ggplot2 3.3.6, knitr 1.39, refelectrodes 0.0.0.9000, tidyr 1.2.1, xtable 1.8-4
- Loaded via a namespace (and not attached): assertthat 0.2.1, cli 3.4.1, colorspace 2.0-3, compiler 4.1.3, DBI 1.1.3, digest 0.6.29, ellipsis 0.3.2, evaluate 0.15, fansi 1.0.3, farver 2.1.1, filehash 2.4-3, generics 0.1.3, glue 1.6.2, grid 4.1.3, gtable 0.3.0, highr 0.9, labeling 0.4.2, lifecycle 1.0.3, magrittr 2.0.3, munsell 0.5.0, pillar 1.8.1, pkgconfig 2.0.3, purrr 0.3.5, R6 2.5.1, rlang 1.0.6, scales 1.2.0, stringi 1.7.8, stringr 1.4.0, tibble 3.1.8, tidyselect 1.2.0, tikzDevice 0.12.3.1, tinytex 0.40, tools 4.1.3, utf8 1.2.2, vctrs 0.5.1, withr 2.5.0, xfun 0.31

```
pdfTeX 3.141592653-2.6-1.40.25 (TeX Live 2023)
kpathsea version 6.3.5
Copyright 2023 Han The Thanh (pdfTeX) et al.
There is NO warranty. Redistribution of this software is
covered by the terms of both the pdfTeX copyright and
the Lesser GNU General Public License.
For more information about these matters, see the file
named COPYING and the pdfTeX source.
Primary author of pdfTeX: Han The Thanh (pdfTeX) et al.
Compiled with libpng 1.6.39; using libpng 1.6.39
Compiled with zlib 1.2.13; using zlib 1.2.13
Compiled with xpdf version 4.04
```