December 31, 2014

0.0.1 convertDocs

`convertDocs` converts between Rmd and Rnw files. The project's `docs/Rmd` or `docs/Rnw` directory is specified. Any files of the same type as the directory are converted to the other type and saved to the other directory. The input files are not removed.

This function speeds up the process of duplicating files, e.g., when wanting to make PDFs from Rnw files when only Rmd files exist. This is almost exclusively what I use this function for. On less frequent occasions I have used it in the other direction when I have Rnw files which were once used to make PDFs but later I decide to put them on the web as a web page and not as a link to a PDF.

The user still makes specific changes by hand, for example, any required changes to `knitr` code chunk options that must differ for PDF output vs. html output. The primary benefit is in not having to fuss with large amounts of standard substitutions which can be automated, such as swapping code chunk enclosure styles and common file metadata. Of course, this function is not perfect. It amounts to a text-parsing hack that is intended to handle the most common of idiosyncrasies and differences which exist between my own Rmd and Rnw files in the context of my own set of rules and assumptions, outlined below.

```
# Rmd <-> Rnw document conversion Conversion support functions Main
# conversion function
convertDocs <- function(path, rmdChunkID = c("```{r", "}", "```"), rnwChunkID = c("<<",
    ">>=", "@"), emphasis = "replace", overwrite = FALSE, ...) {
    stopifnot(is.character(path))
    type <- basename(path)
    rmd.files <- list.files(path, pattern = ".Rmd$", full = TRUE)
    rnw.files <- list.files(path, pattern = ".Rnw$", full = TRUE)
    dots <- list(...)
    if (rmdChunkID[1] == "```{r")
        rmdChunkID[1] <- paste0(rmdChunkID[1], " ")
    gsbraces <- function(txt) gsub("\\{", "\\\\{", txt)
    if (type == "Rmd") {
        stopifnot(length(rmd.files) > 0)
        outDir <- file.path(dirname(path), "Rnw")
        if (is.null(doc.class <- dots$doc.class))
            doc.class <- "article"
        if (is.null(doc.packages <- dots$doc.packages))
            doc.packages <- "geometry"
        doc.class.string <- paste0("\\documentclass{", doc.class, "}")
        doc.packages.string <- paste0(sapply(doc.packages, function(x) paste0("\\usepackage{",
            x, "}")), collapse = "\n")
        if ("geometry" %in% doc.packages)
            doc.packages.string <- c(doc.packages.string, "\\geometry{verbose, tmargin=2.5cm, bmargin=2.
        header.rnw <- c(doc.class.string, doc.packages.string, "\\begin{document}\n")  #,
        # paste0('<<highlight, echo=FALSE>>=\nknit_theme$set(knit_theme$get('',
        # theme, ''))\n@\n'))
    } else if (type == "Rnw") {
        stopifnot(length(rnw.files) > 0)
        outDir <- file.path(dirname(path), "Rmd")
```

```
} else stop("path must end in 'Rmd' or 'Rnw'.")

swapHeadings <- function(from, to, x) {
    nc <- nchar(x)
    ind <- which(substr(x, 1, 8) == "\\section" | substr(x, 1, 4) == "\\sub")
    if (!length(ind)) {
        # assume Rmd file
        ind <- which(substr(x, 1, 1) == "#")
        ind.n <- rep(1, length(ind))
        for (i in 2:6) {
            ind.tmp <- which(substr(x[ind], 1, i) == substr("######", 1,
              i))
            if (length(ind.tmp))
              ind.n[ind.tmp] <- ind.n[ind.tmp] + 1 else break
        }
        for (i in 1:length(ind)) {
            n <- ind.n[i]
            input <- paste0(substr("######", 1, n), " ")
            h <- x[ind[i]]
            h <- gsub("\\*", "_", h)  # Switch any markdown boldface asterisks in headings to double
            heading <- gsub("\n", "", substr(h, n + 2, nc[ind[i]]))
            # h <- gsub(input, '', h)
            if (n <= 2)
              subs <- "\\" else if (n == 3)
              subs <- "\\sub" else if (n == 4)
              subs <- "\\subsub" else if (n >= 5)
              subs <- "\\subsubsub"
            output <- paste0("\\", subs, "section{", heading, "}\n")
            x[ind[i]] <- gsub(h, output, h)
        }
    } else {
        # assume Rnw file
        ind <- which(substr(x, 1, 8) == "\\section")
        if (length(ind)) {
            for (i in 1:length(ind)) {
              h <- x[ind[i]]
              heading <- paste0("## ", substr(h, 10, nchar(h) - 2))
              x[ind[i]] <- gsub(gsbraces(h), heading, h)
            }
        }
        ind <- which(substr(x, 1, 4) == "\\sub")
        if (length(ind)) {
            for (i in 1:length(ind)) {
              h <- x[ind[i]]
              z <- substr(h, 2, 10)
              if (z == "subsubsub") {
                p <- "##### "
                n <- 18
              } else if (substr(z, 1, 6) == "subsub") {
                p <- "#### "
                n <- 15
              } else if (substr(z, 1, 3) == "sub") {
                p <- "### "
```

```
                n <- 12
              }
              heading <- paste0(p, substr(h, n, nchar(h) - 2))
              x[ind[i]] <- gsub(gsbraces(h), heading, h)
            }
        }
    }
    x
}

swapChunks <- function(from, to, x) {
    nc <- nchar(x)
    chunk.start.open <- substr(x, 1, nchar(from[1])) == from[1]
    chunk.start.close <- substr(x, nc - 1 - nchar(from[2]) + 1, nc - 1) ==
        from[2]
    chunk.start <- which(chunk.start.open & chunk.start.close)
    chunk.end <- which(substr(x, 1, nchar(from[3])) == from[3] & nc == nchar(from[3]) +
        1)
    x[chunk.start] <- gsub(from[2], to[2], gsub(gsbraces(from[1]), gsbraces(to[1]),
        x[chunk.start]))
    x[chunk.end] <- gsub(from[3], to[3], x[chunk.end])
    chunklines <- as.numeric(unlist(mapply(seq, chunk.start, chunk.end)))
    list(x, chunklines)
}

# I know I use '**' strictly for bold font in Rmd files.  For now, this
# function assumes: 1. The only emphasis in a doc is boldface or typewriter.
# 2. These instances are always preceded by a space, a carriage return, or
# an open bracket, 3. and followed by a space, period, comma, or closing
# bracket.
swapEmphasis <- function(x, emphasis = "remove", pat.remove = c("`", "\\*\\*",
    "__"), pat.replace = pat.remove, replacement = c("\\\\texttt\\{", "\\\\textbf\\{",
    "\\\\textbf\\{", "\\}", "\\}", "\\}")) {

    stopifnot(emphasis %in% c("remove", "replace"))
    n <- length(pat.replace)
    rep1 <- replacement[1:n]
    rep2 <- replacement[(n + 1):(2 * n)]
    prefix <- c(" ", "^", "\\{", "\\(")
    suffix <- c(" ", ",", "-", "\n", "\\.", "\\}", "\\)")
    n.p <- length(prefix)
    n.s <- length(suffix)
    pat.replace <- c(paste0(rep(prefix, n), rep(pat.replace, each = n.p)),
        paste0(rep(pat.replace, each = n.s), rep(suffix, n)))
    replacement <- c(paste0(rep(gsub("\\^", "", prefix), n), rep(rep1, each = n.p)),
        paste0(rep(rep2, each = n.s), rep(suffix, n)))
    if (emphasis == "remove")
        for (k in 1:length(pat.remove)) x <- sapply(x, function(v, p, r) gsub(p,
            r, v), p = pat.remove[k], r = "")
    if (emphasis == "replace")
        for (k in 1:length(pat.replace)) x <- sapply(x, function(v, p, r) gsub(p,
            r, v), p = pat.replace[k], r = replacement[k])
    x
}
```

```
swap <- function(file, header = NULL, outDir, ...) {
    title <- list(...)$title
    author <- list(...)$author
    highlight <- list(...)$highlight
    ext <- tail(strsplit(file, "\\.")[[1]], 1)
    l <- readLines(file)
    l <- l[substr(l, 1, 7) != "<style>"]  # Strip any html style lines
    if (ext == "Rmd") {
        hl.default <- "solarized-light"
        out.ext <- "Rnw"
        h.ind <- 1:which(l == "---")[2]
        h <- l[h.ind]
        t.ind <- which(substr(h, 1, 7) == "title: ")
        a.ind <- which(substr(h, 1, 8) == "author: ")
        highlight.ind <- which(substr(h, 1, 11) == "highlight: ")
        if (is.null(title) & length(t.ind))
            title <- substr(h[t.ind], 8, nchar(h[t.ind])) else if (is.null(title))
            title <- ""
        if (is.null(author) & length(a.ind))
            author <- substr(h[a.ind], 9, nchar(h[a.ind])) else if (is.null(author))
            author <- ""
        if (is.null(highlight) & length(highlight.ind))
            highlight <- substr(h[highlight.ind], 12, nchar(h[highlight.ind])) else if (is.null(high
            highlight <- hl.default else if (!(highlight %in% knit_theme$get()))
            highlight <- hl.default
        if (!is.null(title))
            header <- c(header, paste0("\\title{", title, "}"))
        if (!is.null(author))
            header <- c(header, paste0("\\author{", author, "}"))
        if (!is.null(title))
            header <- c(header, "\\maketitle\n")
        header <- c(header, paste0("<<highlight, echo=FALSE>>=\nknit_theme$set(knit_theme$get('",
            highlight, "'))\n@\n"))
    } else if (ext == "Rnw") {
        hl.default <- "tango"
        out.ext <- "Rmd"
        begin.doc <- which(l == "\\begin{document}")
        make.title <- which(l == "\\maketitle")
        if (length(make.title))
            h.ind <- 1:make.title else h.ind <- 1:begin.doc
        h <- l[h.ind]
        t.ind <- which(substr(h, 1, 6) == "\\title")
        a.ind <- which(substr(h, 1, 7) == "\\author")
        highlight.ind <- which(substr(l, 1, 11) == "<<highlight")
        if (is.null(title) & length(t.ind))
            title <- substr(h[t.ind], 8, nchar(h[t.ind]) - 1)
        if (is.null(author) & length(a.ind))
            author <- substr(h[a.ind], 9, nchar(h[a.ind]) - 1)
        if (length(highlight.ind)) {
            l1 <- l[highlight.ind + 1]
            h1 <- substr(l1, nchar("knit_theme$set(knit_theme$get('") +
              1, nchar(l1) - nchar("'))\n"))
            if (!(h1 %in% knit_theme$get()))
```

```
                    h1 <- hl.default
            }
            if (is.null(highlight) & length(highlight.ind))
                highlight <- h1 else if (is.null(highlight))
                highlight <- hl.default else if (!(highlight %in% knit_theme$get()))
                highlight <- hl.default
            header <- rmdHeader(title = title, author = author, highlight = highlight)
        }
        header <- paste0(header, collapse = "\n")
        l <- paste0(l[-h.ind], "\n")
        if (ext == "Rmd") {
            from <- rmdChunkID
            to <- rnwChunkID
        }
        if (ext == "Rnw") {
            from <- rnwChunkID
            to <- rmdChunkID
        }
        l <- swapHeadings(from = from, to = to, x = l)
        chunks <- swapChunks(from = from, to = to, x = l)
        l <- chunks[[1]]
        if (ext == "Rmd")
            l <- swapEmphasis(x = l, emphasis = emphasis)
        if (ext == "Rmd")
            l[-chunks[[2]]] <- sapply(l[-chunks[[2]]], function(v, p, r) gsub(p,
                r, v), p = "_", r = "\\\\_")
        l <- c(header, l, "\n\\end{document}\n")
        outfile <- file.path(outDir, gsub(paste0("\\.", ext), paste0("\\.",
            out.ext), basename(file)))
        if (overwrite || !file.exists(outfile)) {
            sink(outfile)
            sapply(l, cat)
            sink()
            print(paste("Writing", outfile))
        }
    }

    if (type == "Rmd") {
        sapply(rmd.files, swap, header = header.rnw, outDir = outDir, ...)
        cat(".Rmd to .Rnw file conversion complete.\n")
    } else {
        sapply(rnw.files, swap, header = NULL, outDir = outDir, ...)
        cat(".Rnw to .Rmd file conversion complete.\n")
    }
}
```