Sometimes datasets refer to the same gene using different names or symbols before they enter the MetaIntegrator pipeline. This can mess up the analysis, for example, by eliminating such genes from consideration from the list of differentially expressed genes because no gene name/symbol/reference appears in enough datasets. Here, different datasets refer to the Septin 9 (name) or SEPT9 (symbol) gene differently:

GSE	Septin 9 in GSE	SEPT9 in GSE	GPL
GSE19491	TRUE	FALSE	GPL6947
GSE28623	FALSE	TRUE	GPL4133
GSE34608	FALSE	TRUE	GPL6480, GPL7731
GSE37250	TRUE	FALSE	GPL10558
GSE39939	TRUE	FALSE	GPL10558
GSE39939.noCultureNeg	TRUE	FALSE	GPL10558
GSE39940	TRUE	FALSE	GPL10558
GSE40553	TRUE	FALSE	GPL10558
GSE42834	TRUE	FALSE	GPL10558
GSE56153	TRUE	FALSE	GPL6883
GSE62147	FALSE	TRUE	GPL6480
GSE74092	FALSE	FALSE	GPL21040
GSE54992	FALSE	TRUE	GPL570
GSE62525	FALSE	TRUE	GPL16951
GSESekaly	FALSE	FALSE	GPL10558
GSEScribaDay0to7	FALSE	FALSE	GPL11154
GSEScribaDay8to180	FALSE	FALSE	GPL11154
GSEScribaDay181to360	FALSE	FALSE	GPL11154
GSEScribaDay541to720	FALSE	FALSE	GPL11154
GSE84076	FALSE	FALSE	GPL16791
GSE83456	FALSE	FALSE	GPL10558
GSE101705	FALSE	TRUE	GPL18573
GSE107731	FALSE	FALSE	GPL15207
GSE81746	FALSE	TRUE	GPL17077
GSE29536_TB	FALSE	FALSE	GPL6102
GSE_MTAB_4257	FALSE	FALSE	A-AGIL-28, A-MEXP-2104
GSE50834	FALSE	FALSE	GPL10558
GSE83892	FALSE	FALSE	GPL10558
GSE73408	FALSE	TRUE	GPL11532
GSE69581	FALSE	FALSE	GPL10558
GSECliff.combined	FALSE	TRUE	NA
GSEScribaDay361to540	FALSE	FALSE	GPL11154

One solution is to map all possible references to a gene to the standardized, unique *gene symbol* as described here, i.e. name deduplication. Aditya Rao's `MetaIntegrator::geneNameCorrection`, Francesco Vallania and Andrew Tam's `MetaIntegrator:::.GEO_fData_key_parser` and the R package HGNCHelper already deduplicate many cases; for example, `MetaIntegrator::geneNameCorrection` deduplicates unusual genes which are often referred to by their names instead of their symbols (e.g. Septin 9 instead of SEPT9) because their symbols will get parsed into dates by Microsoft Excel look like dates. A comprehensive search of genes in our datasets that aren't symbols might reveal more cases of genes that need to be deduplicated. That's what I did below for the genes in `TB_human_datasets_04_2018`:

```
all_genes = purrr::map(TB_human_datasets_04_2018, ~ .$keys) %>% unlist %>% unique
# Follow https://www.genenames.org/about/guidelines#genesymbols to identify invalid symbols
weird_genes = all_genes[which(!grepl("^[A-Z]([,;A-Z0-9-]| /// )*$", all_genes) & !grepl("orf", all_genes) & nchar(all_genes) > 0)]
length(weird_genes)
```

```
## [1] 5920
```

It turns out that almost half of the 5920 are `"HS\.[A-Z0-9]+"` genes that come from just one dataset, GSE83892, so let's ignore those.

```
weird_genes = weird_genes[which(!grepl("^HS\\.", weird_genes))]
length(weird_genes)
```

```
## [1] 2650
```

By continuing to browse `weird_genes`, break it down into cases (e.g. `"HS\.[A-Z0-9]+"`) and understand which cases are not covered by the aforementioned solutions, the reader can identify what remaining logic needs to be written into MetaIntegrator to deduplicate the remaining unsolved cases, which could be corrupting the data in existing analyses. I'm deprioritizing finishing this up because I think we've hit an 80/20 solution for dedpulication, but I could be wrong.