

Mit biber --tool Bibliografieeinträge bearbeiten

Lukas C. Bossert

Es gibt verschiedene Programme, mit denen .bib-Dateien bearbeitet werden können (bspw. JabRef, BibDesk etc.). Möchte man hingegen Bibliografieeinträge in einem automatisierten Prozess verändern, dann bedarf es eines anderen Werkzeugs.

In diesem Beitrag wird gezeigt, wie mit biber im tool-Modus beliebig große .bib-Dateien bearbeitet werden können.

Ausgangspunkt meiner Beschäftigung mit biber und dem Werkzeugmodus war ein konkreter Anlass: Ich brauchte fortlaufend aktualisierte Listen mit Publikationen, die innerhalb eines Sonderforschungsbereichs (SFB) entstanden sind. Da eine händische Pflege dieser Liste mir deutlich zu aufwendig schien und die Daten über die Hochschulbibliografie der Bibliothek ohnehin gepflegt werden,¹ wollte ich einen effizienteren Weg gehen.²

Fast alle Bibliothekskataloge bieten die Möglichkeit an, Bibliografieeinträge im BibTeX-Format zu exportieren. Dabei sind die Publikationsmetadaten sehr unterschiedlich was bspw. die Ausführlichkeit der Beschreibung oder deren Ausgabe angeht.³ Eine Nachbearbeitung der Metadaten ist daher von Fall zu Fall notwendig. Diese wollte ich aber so automatisiert wie möglich gestalten. Genau dafür eignet sich biber hervorragend.

Ausgangslage

Dank einer API-gestützten Abfrage konnte ich mir über die Kommandozeile eine .bib-Datei mit allen Publikationen des SFBs herunterladen und als sfb.bib speichern.

```
curl "http://publications.rwth-aachen.de/PubExporter.py?p=pid:G:(GEPRIS)
↪403224013&of=hx" > sfb.bib
```

Ein willkürlich herausgegriffener Datensatz sieht so aus:

¹ <https://publications.rwth-aachen.de/record/783071>

² Die nachfolgenden Beispiele sind zwar konkrete BibTeX-Einträge, die Vorgehensweise mit biber ist jedoch exemplarisch zu verstehen.

³ Oft werden noch "veraltete" Feldnamen wie address oder journal exportiert.

```

1 % IMPORTANT: The following is UTF-8 encoded. This means that in the presence
2 % of non-ASCII characters, it will not work with BibTeX 0.99 or older.
3 % Instead, you should use an up-to-date BibTeX implementation like "bibtex"8 or
4 % "biber".
5
6 @ARTICLE{Babbar:782496,
7   author      = {Babbar, Anshu and Hitch, Thomas C. A. and others},
8   title       = {{T}he {C}ompromised {M}ucosal {I}mmune {S}ystem of  $\beta$ 7
9                 {I}ntegrin-{D}eficient {M}ice {H}as {O}nly {M}inor {E}
10    ↪ffects on the {F}ecal {M}icrobiota in {H}omeostasis},
11   journal     = {Frontiers in microbiology},
12   volume      = {10},
13   issn        = {1664-302X},
14   address     = {Lausanne},
15   publisher   = {Frontiers Media},
16   reportid    = {RWTH-2020-01913},
17   pages       = {2284},
18   year        = {2019},
19   cin         = {526000-2 / 525500-2 / 537500-2},
20   ddc         = {570},
21   cid         = {$I:(DE-82)526000-2_20140620$ / $I:(DE-82)525500-2_20140620$ / $
22    ↪I:(DE-82)537500-2_20140620$},
23   pnm         = {SFB 1382 - SFB 1382: Die Darm-Leber-Achse - Funktionelle
24    ↪Zusammenhänge und therapeutische Strategien (403224013)},
25   pid         = {G:(GEPRIS)403224013},
26   typ         = {PUB:(DE-HGF)16},
27   UT          = {WOS:000491333100001},
28   doi         = {10.3389/fmicb.2019.02284},
29   url         = {https://publications.rwth-aachen.de/record/782496},
30 }

```

Neben den bereits genannten “veralteten” Feldnamen, wie `address` oder `journal` gibt es noch weitere Feldnamen, die nicht zum BibLaTeX-Format gehören: `reportid`, `cin`, `ddc`, `cid`, `pnm`, `pid`, `UT`. Diese Felder werden von BibLaTeX standardmäßig nicht verarbeitet und der gewählte Bibliografiestil müsste diese explizit in ein eigenes Datenmodell aufnehmen.⁴ Zudem werden alle Großbuchstaben im Titel mit `{ }` eingefasst, um eine Kleinsetzung zu vermeiden. Diese Großschreibug (Capitalizing) sollte aber nicht aus dem Datensatz heraus erzwungen, sondern vom Bibliografiestil umgesetzt werden.

⁴ Da es sich bei diesen Feldern um interne Informationen handelt, die im Normalfall ohnehin von BibLaTeX ignoriert werden, ist es nur mein eigener ästhetischer Anspruch an die BibTeX-Darstellung, diese Felder zu löschen.

Die Bearbeitung des Datensatzes mit biber kümmert sich ohne besondere Spezifikation um mehrere Punkte. Mit ein paar Optionen lassen sich weitere Anpassungen vornehmen:

- `--tool` startet biber im Werkzeugmodus.
- Mit `--outfile <FILE.bib>` wird eine explizite Ausgabedatei angegeben. In unserem Fall ist es derselbe Name der Eingabedatei; ansonsten wird biber eine `.bib`-Datei mit dem Zusatz `_bibertool` erstellen.
- `--output_align` sorgt für eine ausgerichtete Darstellung der Feldnamen und -inhalte.
- `--output_indent` regelt, um wie viele Leerzeichen eingerückt werden soll.
- Über `--output_fieldcase=lower` wird angegeben, dass alle Feldnamen und der Datensatztyp kleingeschrieben werden.
- Über `--output-field-order=<FELDNAMEN>` kann man die interne Reihenfolge der Felder festlegen.

Am Ende des Befehls steht der Dateiname der zu bearbeitenden `.bib`-Datei.

```
biber --tool --outfile sfb.bib --output_align --output_indent=1 --
  ↪output_fieldcase=lower --output-field-order=author,title,date sfb.bib
```

Das Ergebnis sieht für den Datensatz wie folgt aus:

```
1 @article{Babbar:782496,
2   author      = {Babbar, Anshu and Hitch, Thomas C. A. and others},
3   title       = {{T}he {C}ompromised {M}ucosal {I}mmune {S}ystem of  $\beta$ 7 {I}
  ↪ntegrin-{D}eficient {M}ice {H}as {O}nly {M}inor {E}ffects on the {F}ecal {M}
  ↪icrobiota in {H}omeostasis},
4   date        = {2019},
5   doi         = {10.3389/fmicb.2019.02284},
6   issn        = {1664-302X},
7   journaltitle = {Frontiers in microbiology},
8   location    = {Lausanne},
9   pages       = {2284},
10  publisher    = {Frontiers Media},
11  url          = {https://publications.rwth-aachen.de/record/782496},
12  volume       = {10},
13 }
```

Alle BibLaTeX nicht bekannten Feldnamen werden mit ihren Inhalten gelöscht. Die “veralteten” Feldnamen werden umbenannt (zu `location` und `journaltitle`). Und der Kommentar über dem Eintrag wird ebenfalls entfernt. Die ersten drei Felder werden wie gewünscht sortiert: `author`, `title`, `date`, der Rest wird alphabetisch angeordnet.

Eigene Modifikationen einbauen

Der Datensatz ist an sich BibLaTeX-kompatibel, aber in meinem konkreten Fall mussten zusätzlich mehrere Punkte an die eigenen Bedürfnisse und Vorgaben angepasst werden:

- Die Einklammerung der Großbuchstaben im Titel störte noch beim Lesen;
- Das Feld issn war für alle Datensatztypen nicht notwendig;
- Die Ortsangabe (location) sollte nur für @article gelöscht werden;
- Für interne Zwecke musste das Feld reportid jedoch erhalten bleiben.
- Wenn im Feld prn die DFG-Fördernummer geschrieben steht, musste
 - in das Feld keywords *CRC1382* nachgetragen und
 - der Name des SFBs in das Feld note werden.

Diese weiteren Bearbeitungswünsche sind an sich problemlos umzusetzen, müssen jedoch in einer extern gespeicherten Konfigurationsdatei definiert werden.

Die Konfigurationsdatei (sfb-configuration.conf) basiert auf xml. Das Grundgerüst, das für die Anpassungen notwendig war, sieht so aus:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <config>
3   ... <!-- Anpassungen -->
4   <sourcemap>
5   ... <!-- Anpassungen -->
6   </sourcemap>
7   <datamodel>
8     <fields>
9     ... <!-- Anpassungen -->
10    </fields>
11    <entryfields>
12    ... <!-- Anpassungen -->
13    </entryfields>
14  </datamodel>
15 </config>

```

Da mit weiteren Anpassungen ohnehin eine Konfigurationsdatei benötigt wurde, konnten einige Anweisungen, die ansonsten in der Kommandozeile eingegeben werden, ausgelagert werden. Diese werden direkt unterhalb des Tags <config> eingegeben:

```

<output_fieldcase>lower</output_fieldcase>
<output_indent>1</output_indent>
<output_align>true</output_align>

```

Die nächste Anpassung erfolgt in der Umgebung `<sourcemap>`. In `<sourcemap>` wird, vereinfacht gesagt, ein “Suchen-Ersetzen”-Prinzip (nach bestimmten Bedingungen) ausgeführt. In der Standardkonfigurationsdatei⁵ gibt es ebenfalls eine `<sourcemap>`-Umgebung, die unter anderem die Umbenennung der “veralteten” Feldnamen auslöst. Damit diese wichtige Anpassung nicht verloren geht, müssen alle Zeilen aus dieser originalen `<sourcemap>`-Umgebung⁶ in die eigene Konfigurationsdatei kopiert werden:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <config>
3   <output_fieldcase>lower</output_fieldcase>
4   <output_indent>2</output_indent>
5   <output_align>true</output_align>
6   <sourcemap>
7     <!-- original sourcemap from biber-tool.conf -->
8     <maps datatype="bibtex">
9       <map>
10        <map_step map_type_source="conference" map_type_target="inproceedings"/>
11        <map_step map_type_source="electronic" map_type_target="online"/>
12        <map_step map_type_source="www" map_type_target="online"/>
13      </map>
14      <map map_overwrite="1">
15        <map_step map_final="1" map_type_source="mastersthesis" map_type_target="
16        ↪thesis"/>
17      </map>
18      <map map_overwrite="1">
19        <map_step map_final="1" map_type_source="phdthesis" map_type_target="thesis"/
20        ↪>
21        <map_step map_field_set="type" map_field_value="phdthesis"/>
22      </map>
23      <map map_overwrite="1">
24        <map_step map_final="1" map_type_source="techreport" map_type_target="report"
25        ↪>
26        <map_step map_field_set="type" map_field_value="techreport"/>
27      </map>
28      <map>
29        <map_step map_field_source="address" map_field_target="location"/>
30        <map_step map_field_source="school" map_field_target="institution"/>
31        <map_step map_field_source="annote" map_field_target="annotation"/>

```

⁵ <https://github.com/plk/biber/blob/master/data/biber-tool.conf>

⁶ <https://github.com/plk/biber/blob/b5567eb70e86c22bbfb7696612fd9ec2146eb2/data/biber-tool.conf#L321-L352>

```

30 <map_step map_field_source="journal" map_field_target="journaltitle"/>
31 <map_step map_field_source="archiveprefix" map_field_target="eprinttype"/>
32 <map_step map_field_source="primaryclass" map_field_target="eprintclass"/>
33 <map_step map_field_source="key" map_field_target="sortkey"/>
34 <map_step map_field_source="pdf" map_field_target="file"/>
35 </map>
36 </maps>
37 </sourcemap>
38 <datamodel>
39 <fields>
40 ... <!-- Anpassungen -->
41 </fields>
42 <entryfields>
43 ... <!-- Anpassungen -->
44 </entryfields>
45 </datamodel>
46 </config>

```

In `<sourcemap>` werden die Anpassungen in einzelnen Schritten nacheinander ausgeführt (`<map_step ... />`) und zu größeren Einheiten gebündelt (`<map>...</map>` bzw. `<maps>...</maps>`).

An der Umbenennung von `address` zu `location` sieht man die Funktionsweise.

```
<map_step map_field_source="address" map_field_target="location"/>
```

In einem Bearbeitungsschritt wird innerhalb eines Datensatzes (`@article`, `@book` etc.) nach dem Feldnamen `address` gesucht (`map_field_source`) und durch den Inhalt von `map_field_target` ersetzt (`location`).

Auf diese Weise lassen sich Feldnamen ändern. Feldinhalte, wie bspw. Titel, werden nachfolgend am Beispiel der erzwungenen Großschreibung gezeigt.

```
<map_step map_field_source="title" map_match="{{[A-Z]}}" map_replace="$1"/>
```

Es wird ausschließlich das Feld `title` berücksichtigt (`map_field_source`). Darin wird nach einem Großbuchstaben innerhalb eines geschweiften Klammerpaares gesucht (`map_match="{{[A-Z]}}"`). Für die Suche können Reguläre Ausdrücke verwendet werden,⁷ wie es auch mit `[A-Z]` (jeder Großbuchstabe von A bis Z) gemacht wird. Der Inhalt vom Klammerpaar `()` wird zwischengespeichert und bei `map_replace` mit `$1` wieder ausgegeben.

In einem nächsten Schritt soll für alle `@article`-Einträge das Feld `location` gelöscht werden:

⁷ https://de.wikipedia.org/wiki/Regulärer_Ausdruck

```
<per_type>article</per_type>
<map_step map_field_set="location" map_null="1"/>
```

Über `<per_type><Datensatztyp></per_type>` wird festgelegt, dass die nachfolgenden Schritte nur für den gewählten Datensatztyp `article` gelten. Mit `map_null="1"` wird das Feld `location` gelöscht. Ganz analog wird auch das Feld `issn` gelöscht, nur dass in diesem Schritt keine Einschränkung hinsichtlich des Datensatztyps erfolgt.

Was jetzt noch fehlt, sind die Anpassungen, die nur bei bestimmten Kriterien ausgeführt werden dürfen. Zunächst muss geprüft werden, ob im Feld `pnm` die DFG-Fördernummer (hier: 403224013) steht.

```
<map_step map_field_source="pnm" map_final="1" map_match="\(403224013\)"/>
```

Wenn diese Bedingung nicht erfüllt ist, dann werden wegen `map_final="1"` die nachfolgenden Schritte nicht ausgeführt. Wenn die Bedingung jedoch erfüllt ist, dann soll zunächst ein Keyword gesetzt

```
<map_step map_append="1" map_field_set="keywords" map_field_value=", CRC1382"/>
```

und anschließend der vollständige Name des SFBs in das Feld `note` geschrieben werden.

```
<map_step map_field_set="note" map_field_value="SFB 1382: Die Darm-Leber-Achse --
  ↳ Funktionelle Zusammenhänge und therapeutische Strategien (403224013)"/>
```

Man kann also durchaus auch komplexere Anpassungen vornehmen, die auf bedingten Anweisungen beruhen.

Der Block mit den eigenen Anpassungen sieht insgesamt wie folgt aus:⁸

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <config>
3   ...
4   <sourcemap>
5     <!-- original sourcemap from biber-tool.conf -->
6     <maps datatype="bibtex">
7       ...
8     </maps>
9     <!-- additional configurations -->
10    <maps datatype="bibtex">
11      <map>
12        <!-- In the title get rid off curly brackets -->
13        <map_step map_field_source="title" map_match="\{([A-Z])\}" map_replace="$1"/>
```

⁸ Weitere Beispiele gibt es in Kapitel 3.13 in der Dokumentation von [1].

```

14 </map>
15 <map>
16 <!-- delete 'location' but only for @article -->
17 <per_type>article</per_type>
18 <map_step map_field_set="location" map_null="1"/>
19 </map>
20 <map>
21 <!-- drop these fields -->
22 <map_step map_field_set="issn" map_null="1"/>
23 </map>
24 <map map_overwrite="1">
25 <!-- execute steps only if first step is true -->
26 <map_step map_field_source="prn" map_final="1" map_match="\{403224013\}"/>
27 <!-- append 'CRC1382' to 'keywords' -->
28 <map_step map_append="1" map_field_set="keywords" map_field_value=", CRC1382"
  ↪>/>
29 <!-- write SFB name into field 'note' -->
30 <map_step map_field_set="note" map_field_value="SFB 1382: Die Darm-Leber-
  ↪Achse -- Funktionelle Zusammenhänge und therapeutische Strategien
  ↪(403224013)"/>
31 </map>
32 </maps>
33 </sourcemap>
34 <datamodel>
35 <fields>
36 ... <!-- Anpassungen -->
37 </fields>
38 <entryfields>
39 ... <!-- Anpassungen -->
40 </entryfields>
41 </datamodel>
42 </config>

```

Die letzten beiden notwendigen Anpassungen betreffen das Feld `reportid`, das erhalten bleiben soll. Hierfür wird zuerst in der `<fields>`-Umgebung das Feld an sich definiert.

```
<field datatype="literal" fieldtype="field">reportid</field>
```

In der `<entryfields>`-Umgebung wird nochmals der Feldname aufgeführt:

```
<field>reportid</field>
```

Wiederum im Gesamtzusammenhang:


```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <config>
3     ...
4     <sourcemap>
5         ...
6     </sourcemap>
7     <datamodel>
8         <fields>
9             <!-- do not drop these fields -->
10            <field datatype="literal" fieldtype="field">reportid</field>
11        </fields>
12        <entryfields>
13            <!-- do not drop these fields -->
14            <field>reportid</field>
15        </entryfields>
16    </datamodel>
17</config>

```

Der Befehl für die Kommandozeile muss modifiziert werden, indem mit `--configfile=sfb-configuration.conf` die Konfigurationsdatei geladen wird.

```

biber --tool --outfile sfb.bib --output-field-order=author,title,date --
↳configfile=sfb-configuration.conf sfb.bib

```

Der Eintrag ist nun entsprechend gesäubert und nach den eigenen Wünschen angepasst:

```

1 @article{Babbar:782496,
2   author    = {Babbar, Anshu and Hitch, Thomas C. A. and others},
3   title     = {The Compromised Mucosal Immune System of  $\beta$ 7 Integrin-
4   ↳Deficient Mice Has Only Minor Effects on the Fecal Microbiota in Homeostasis
5   ↳},
6   date      = {2019},
7   doi       = {10.3389/fmicb.2019.02284},
8   journaltitle = {Frontiers in microbiology},
9   keywords  = {,CRC1382},
10  note      = {SFB 1382: Die Darm-Leber-Achse -- Funktionelle Zusammenhänge
11  ↳und therapeutische Strategien (403224013)},
12  pages     = {2284},
13  publisher  = {Frontiers Media},
14  reportid   = {RWTH-2020-01913},
15  url       = {https://publications.rwth-aachen.de/record/782496},
16  volume    = {10},
17 }

```

Fazit

Es wurde gezeigt, wie man über den Werkzeugmodus von biber beliebig viele Bibliografieeinträge nach eigenen Wünschen und Vorgaben bearbeiten kann. Dabei lässt sich der Prozess über die Kommandozeile steuern, sodass dies in einen vollautomatisierten Ablauf eingebunden werden kann, der sich vom Abrufen der Daten aus einem OPAC, über die Bereinigung und Anpassung bis zur Erstellung von Publikationslisten kümmert und darüber hinaus die Daten auf einen Server lädt, um sie mit einer Instanz von *bibtex-browser*⁹ darzustellen.

Acknowledgments

“DFG (German Research Foundation) – Project-ID 403224013 – SFB 1382”

Literatur und Software

- [1] Philip Kime, François Charette: The Biber package, A BibTeX replacement for users of BibLaTeX, Version 2.14, <http://biblatex-biber.sourceforge.net/> (besucht am 11. 3. 2020).

⁹<https://www.monperrus.net/martin/bibtexbrowser/>