

Do not change this document

User Manual for the BELS Georeference Matcher

What is the BELS Georeference Matcher?	1
The BELS Georeference Matching Process	2
Georeference Matcher Web App	4
Input File Preparation	4
File Format	5
File Size	5
Field Names	5
Field Name Modifications	5
User Interface	6
Step 1: File Selection	6
Step 2: Notification destination	6
Step 3: Output file name	7
Step 4: Submit	7
Georeference Matcher Web App Output	8
Opening the Output File	8
Fields in the Output File	8
Example	8
Georeference Matcher API	10
Georeference Matcher API Output	11
Georeference Matcher Field Descriptions	11

What is the BELS Georeference Matcher?

The Georeference Matcher is one service among the Biodiversity Enhanced Location Services, also known as BELS (<https://github.com/VertNet/bels>). The Georeference Matcher tries to find best possible georeferences to match input data interpretable as Darwin Core Location terms (<https://dwc.tdwg.org/terms/#location>) from among all georeferences shared via the Global Biodiversity Information Facility (GBIF - <https://www.gbif.org/>), iDigBio (<https://www.idigbio.org/portal/search>), and the VertNet gazetteer of georeferences assembled from the large-scale collaborative georeferencing projects MaNIS, HerpNET, and ORNIS.

The Georeference Matcher can be used via a web application (temporarily at <https://localityservice.uc.r.appspot.com/>) or via an Application Programming Interface (API - via http POST requests to <https://localityservice.uc.r.appspot.com/api/bestgeoref>). The web application is the preferred method for bulk georeferencing up to about 100k records (actual number determined by a 32MB limit on the content of the CSV file being uploaded), while the API is best for integration into systems that require georeference matches for records one at a time. The web application can be expected to provide a CSV file of the original input with additional fields populated by matched georeference data within a few minutes. The API can be expected to return the result of the matching process for a single record within a few seconds.

The BELS Georeference Matching Process

The BELS Georeference Matcher takes advantage of a gazetteer containing georeferences constructed from nearly 156M distinct Locations from the sources described in the preceding section. The entire gazetteer is preprocessed to enable quick matching of relevant content of Darwin Core Location fields. The gazetteer does this by creating three distinct strings to use to find a match. The web app and API then run the same process on the input Location fields to construct three matching strings. The Georeference Matcher then attempts to find a match using these strings in a priority order to find the best possible georeference. "Best" here refers to "geographic coordinates plus a coordinate reference system plus an uncertainty or a spatial geometry with a coordinate reference system" as defined in Chapman and Wieczorek (2020). If a complete (best) georeference isn't found, the Georeference Matcher returns the best matching geographic coordinates, if they can be found. The characteristics and preparation of the gazetteer is the subject of a manuscript (Zermoglio et al., in prep). The process used by the Georeference Matcher to take advantage of the gazetteer is given briefly here in the following steps:

Step 1:

For each record, the Georeference Matcher first translates the field names as described in the section **Field Name Modifications**.

Step 2:

From the fields whose modified names are Darwin Core Location class terms, the Georeference Matcher assesses whether there is at least one field that can be used to determine an ISO 2-letter country code. Such a field can have blank values, but the field must be in the input. If not, an error is generated for the user and processing ends.

Step 3:

If the input is processed successfully, the user is alerted that a notification with a download link will be sent to the email address provided.

Step 4:

A field from which to establish an ISO 2-letter country code is determined using the following priority order: *countryCode*, *country*. The new field *match_country* is created to store these data.

Step 5:

A value for *interpreted_countrycode* is ascertained from the *countrycode_lookup* vocabulary (the source for which can be found at https://github.com/VertNet/bels/blob/main/data/countrycode_lookup.csv) using the value in *match_country*.

Step 6:

Three distinct matching strings are created for the purpose of finding georeferences for the Location in the input data. The strings are meant to create matches for three cases by concatenation of the values of specific Location fields.

- **match sans coordinates** - the matching string uses content from the following fields after interpretation as Darwin Core (the order does not matter in the input file):
 - waterbody, islandGroup, island, interpreted_countrycode* (from Step 5), *stateProvince, county, municipality, locality, verbatimLocality, minimumElevationInMeters, maximumElevationInMeters, verbatimElevation, verticalDatum, minimumDepthInMeters, maximumDepthInMeters, verbatimDepth*.
- **match using verbatim coordinates** - the matching string uses the content of the fields listed under **match sans coordinates**, above, plus the following fields after interpretation as Darwin Core:
 - verbatimCoordinates, verbatimLatitude, verbatimLongitude*
- **match using all coordinates** - the matching string uses the content of the fields listed under **match sans coordinates** plus those **under match using verbatim coordinates**, above, plus the following fields after interpretation as Darwin Core:
 - decimalLatitude, decimalLongitude*

Several transformations occur during the process of converting the content of the fields into the final matching strings. First, note that the content of the continent field is omitted. Research showed that the inclusion of *continent* more often than not resulted in fewer matches in practice without resulting in spurious matches. Second, where the uppercase values of the fields *locality* and *verbatimLocality* are the same, only one copy was used. This is to group Locations that have only one of the two fields in the content with those that have both, and are effectively the same. Third, the numeric values of the fields *decimalLatitude* and *decimalLongitude* are rounded to seven decimal places and cast as strings. This interpretation is done to avoid spurious distinct Locations due to coordinate precision issues in the original data.

With these three adjustments in place, four additional modifications are made to create the final matching strings. First, for each type of matching string, the contents of all of the

fields mentioned above are concatenated in the order given. Second, characters with diacritics are simplified to their ASCII root characters (e.g., 'ç' would become 'c', 'é' would become 'e'). Third, all instances of symbols, except those that contribute to the proper interpretation of numbers (e.g., '-', '.', '/'), are removed. Fourth, the resulting strings up to this point are unicode normalized using the KFKC normal form (Unicode characters are standardized to canonical values) and case folded (made all lower case). Finally, characters not used for the semantics of numbers (e.g., '.' not between digits) are removed.

Step 7:

With the matching string determined, the Georeference Matcher begins a process to determine the best georeference matching the input Location data of each input record. If the record already has a georeference with the characteristics of one done using best practices (Chapman & Wieczorek 2020), the georeference is retained. The specific characteristics sought are viable decimalLatitude and decimalLongitude, a realistic coordinateUncertaintyInMeters, and a coordinate reference system (or geodetic datum is sufficient, given that the coordinate format is decimal geographic coordinates) that is unambiguously specified.

For the remaining records, an attempt is made to find the best georeference from the gazetteer using the string described above for **match using all coordinates**. Best georeferences for each match string type are pre-determined and stored in the gazetteer through a complex process (manuscript in preparation).

For records for which a match could not be found using the coordinates, an attempt is made to find the best georeference from the gazetteer using the string described above for **match using verbatim coordinates**. If that fails, the final matching string, **match sans coordinates**, is used to find a best georeference from the gazetteer.

If no match is found using the methods described above, no results will be returned in the fields appended in the output file.

Georeference Matcher Web App

The BELS Georeference Matcher User Interface (<https://localityservice.uc.r.appspot.com/>) requires four simple steps to engage. Once a data file has been prepared, users can upload and submit them to the Georeference Matcher Web App. A link to results in a gzipped CSV file will be sent to the email address requested.

Input File Preparation

To use the Georeference Matcher Web App, users must upload a data file into the user interface. Files must adhere to a set of formats and preparations. Files that do not meet these

requirements may not be accepted by the Georeference Matcher or may return outputs with errors and other incorrect data.

File Format

All files uploaded to the Georeference Matcher must be a comma-separated text file (CSV). Other file formats may be accepted in future versions. Files that are not comma-separated will not be accepted by the web app.

Files encoded as UTF-8 are strongly recommended, but not required.

File Size

Files uploaded to the web app must not exceed 32Mb. The more content there is in the input fields, the fewer the records that can fit in the 32 MB limit and the longer the file requires for uploading. A good target maximum dataset is about 100k records. Aside from the limitation just described and the upload time (which is subject to local connectivity), the size of the file does not have a great effect on the processing time. Most files should be processed and an email sent to the address provided within a minute.

Field Names

The Georeference Matcher will accept any number of fields in the input data file, but the file must have a header row with the names of the fields, and every row must have the same number of fields as the header. All fields will be retained, but field names may be modified to comply with certain limitations during processing. The modified fields will be present in the output file. The Georeference Matcher will use fields recognized as Darwin Core Location terms (<https://dwc.tdwg.org/terms/#location>) to attempt to make matches with existing records and find the best possible existing georeference. The complete list of the field names that Georeference Matcher can interpret can be found at (https://github.com/VertNet/bels/blob/main/bels/vocabularies/darwin_cloud.txt).

If you have unambiguous alternative field names that you would like to see added to this list, please suggest them via a new issue in the BELS GitHub repository (<https://github.com/VertNet/bels/issues>).

Field Name Modifications

Field names that ...

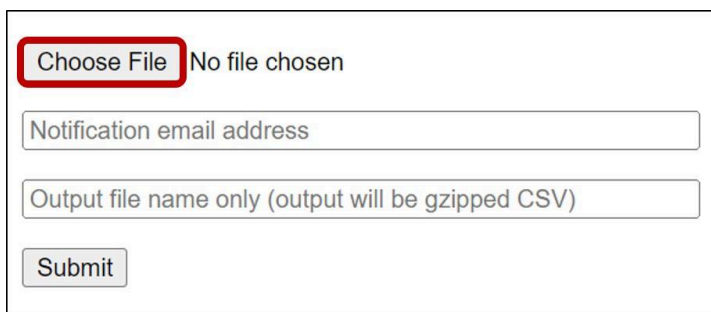
- can be interpreted unambiguously to be the same as a Darwin Core term name will be changed to that term name in lower case (e.g., "*verbatimcoordinates*" will be substituted for "*COORDINATES*");
- have uppercase letters will be converted to all lowercase (e.g., "*stateprovince*" will be substituted for "*stateProvince*");

- have white space characters at the beginning or end will have those characters trimmed (e.g., "country" will be substituted for "country");
- begin with a number will have an underscore inserted before the number (e.g., "_1adminBoundary" will be substituted for "_1adminBoundary");
- contain characters other numbers, letters or underscores (e.g., #, \$, ?, *, -, /, etc.) will be replaced with an underscore, "_"; if this occurs at the beginning of the field name, two underscores will be substituted (e.g., "__catnum" will be substituted for "#catnum");
- contain spaces anywhere except the beginning or end will have each such space replaced by an underscore (e.g., "catalog_number" will be substituted for "CatalogNumber")
- are blank after trimming will be named following the pattern "unnamed_column_n" where n is the least unused ordinal number among field names (i.e., the first blank field name, after trimming, would be "unnamed_column_1", the second, "unnamed_column_2", etc.)
- appear more than once in the file will have the second and subsequent instances prepended with a number of underscores equal to the number of times the field name appeared on the header before this instance of the same field name (e.g., with the fields "country", "Country" and "COUNTRY" in the header, in that order, the first would be remain as "country", the second would become "_country", and the third "__country")
- after all of the above, exceed 128 characters in length will be truncated to 128 characters.

User Interface

Step 1: File Selection

The Georeference Matcher user interface is simple and direct. The first step for data users is to select the data file they wish to upload.



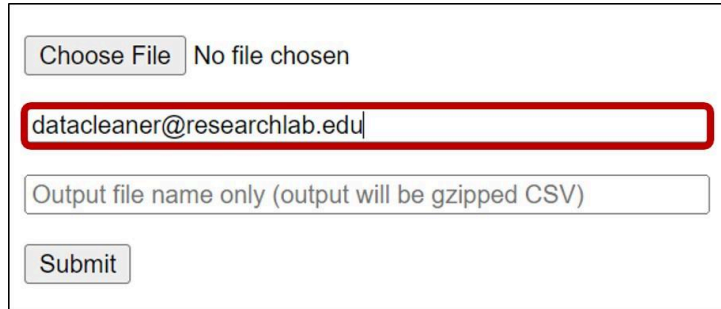
The screenshot shows a web form with a red border. At the top, there is a button labeled "Choose File" with a red border, followed by the text "No file chosen". Below this is a text input field labeled "Notification email address". Underneath that is another text input field labeled "Output file name only (output will be gzipped CSV)". At the bottom left of the form is a "Submit" button.

Click the **Choose File** button. A file selection window will open to permit the selection of a data file on a local computer or network.

Select the desired data file. Click **Open**.

Step 2: Notification destination

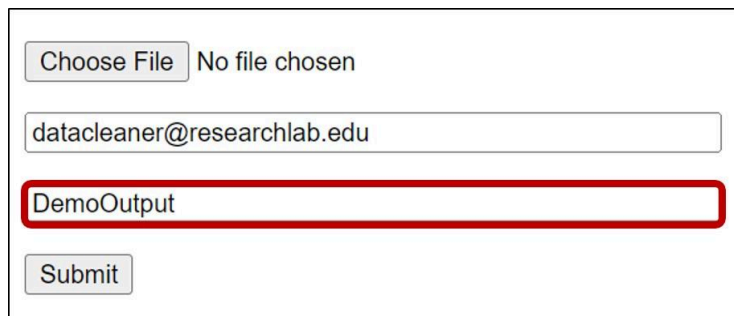
Next, enter the email address to which the service should return the link to the output file. Currently, only one email address is permitted for each submission.



A screenshot of a web form for Step 2. At the top, there is a button labeled 'Choose File' followed by the text 'No file chosen'. Below this is a text input field containing the email address 'datacleaner@researchlab.edu', which is highlighted with a red rectangular border. Underneath the email field is another text input field with the placeholder text 'Output file name only (output will be gzipped CSV)'. At the bottom of the form is a button labeled 'Submit'.

Step 3: Output file name

Provide a file name for the output file.



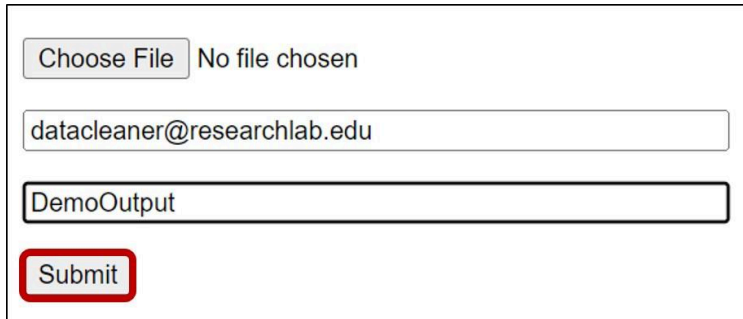
A screenshot of a web form for Step 3. It features a 'Choose File' button and 'No file chosen' text at the top. The email address 'datacleaner@researchlab.edu' is entered in the top text field. The bottom text field contains the filename 'DemoOutput' and is highlighted with a red rectangular border. A 'Submit' button is located at the bottom of the form.

When the Georeference Matcher returns a link to an output file to the email address provided in Step 2, the output file name will have a UUID appended to it to distinguish it from any other requested filename that might be otherwise the same. For example, if the name “architeuthis_sightings” is provided, the results will be returned in a zipped CSV (.csv.gz) with this name plus the unique identifier provided by process that ran on that input file.

Example: the requested file named “architeuthis_sightings” will be returned as something akin to “architeuthis_sightings-89b1b305-e10c-4c85-97b3-c25dc32a7f68.csv.gz”.

Step 4: Submit

Finally, click the **Submit** button.



When these steps are completed correctly, a short message will be displayed in the browser window:

“An email with a link to the results will be sent to [email address provided].”

To submit a new data file to the Georeference Matcher, return to <https://localityservice.uc.r.appspot.com/>, or click the **Back** button on your browser and fill in the page with the updated information for the new job.

Georeference Matcher Web App Output

After the Georeference Matcher has processed the submitted data file, an email will be sent to the address provided in Step 2 above (in the section [Georeference Matcher Web App](#)) with a link to download the results. The output file is a CSV file that has been gzipped. When an output file is received, it can be opened using extraction software, such as 7zip, The Unarchiver, or RAR Extractor (applications will vary depending upon the operating system in use on the local computer). Once extracted the output file will be available as a CSV.

Opening the Output File

The output file can be opened by a wide range of text readers and spreadsheet applications. Text readers, such as BBEdit, Atom, Sublime, or Notepad++ are the recommended tools to review the data file because they will not modify the data within. If it is necessary to view the data file in a spreadsheet application, such as Excel or Numbers, **DO NOT DOUBLE CLICK** the CSV file to open it. Doing so is very likely to cause the default spreadsheet application to re-interpret the data in the data file. This may cause changes to the data values produced by the Georeference Matcher and cause any future analyses to be inaccurate and incorrect. Instead, use the spreadsheet application's import procedures to avoid these issues. The file should be imported as a comma delimited file with UTF-8 encoding.

Fields in the Output File

Once open in a text reader or spreadsheet application, interpreted versions (see the section [Field Name Modifications](#), above) of all of the fields that were present in the original data file uploaded to the web app will be present. In addition to the original fields, 18 new columns will have been appended after the last original column. These fields are described at the end of this document in the section [Georeference Matcher Field Descriptions](#).

Example

The uploaded data file contains one record with the following verbatim fields:

CONTINENT: Europe
COUNTRY: Ireland
COUNTRYCODE: IE
LOCALITY: Bunowen R.(at Glengowla)

Step 1:

The field names are translated to *continent*, *country*, *countryCode*, *locality*.

Step 2:

The input contains a *countryCode* field from which the ISO country code can be determined, no error.

Step 3:

The user is notified that an email will be sent to the address provided.

Step 4:

A new field, *match_country*, is created and populated with the value 'IE' from the input field *COUNTRYCODE*.

Step 5:

A new field, *interpreted_countrycode*, is created and populated with the value 'IE' from the preferred value of 'IE' in the *countrycode_lookup* vocabulary.

Step 6:

Three strings to match against are created from the input data, as follow:

match sans coordinates - "iebunowenratglengowla"
match using verbatim coordinates - "iebunowenratglengowla"
match using all coordinates - "iebunowenratglengowla"

Step 7:

The service looks for a match first using the **match using all coordinates** string and, in this example, is unsuccessful. Continuing, the service finds a match using the **match**

using **verbatim coordinates** string. The results of the best georeference for this matching string are:

bels_decimallatitude: 53.065133
bels_decimallongitude: -9.366403
bels_geodeticdatum: epsg:4326
bels_coordinateuncertaintyinmeters: 100

The matching georeference did not contain data in the following fields, so they were left blank:

bels_georeferencedby:
bels_georeferenceddate:
bels_georeferenceprotocol:
bels_georeferencesources:

The matching georeference contained data in georeferenceRemarks and that data was returned:

bels_georeferenceremarks: Data has been captured using a grid reference system. Provided location represents cell's center point and uncertainty represents cell's size

The matching georeference was given a score of "1" because it contained georeferencingRemarks. See the BELS GitHub repository for a full scoring system (<https://github.com/VertNet/bels>):

bels_georeference_score: 1

The source of the georeference was a snapshot of data aggregated via GBIF:

bels_georeference_source: GBIF

The number of matching georeferences was "1".

bels_best_of_n_georeferences: 1

The string for which a match was found is given in the **bels_match_type**:

bels_match_type: match using verbatim coords

Georeference Matcher API

The BELS Georeference Matcher Application Programming Interface (root URL at <https://localityservice.uc.r.appspot.com/api/>) provides the service to find existing georeferences from the gazetteer aggregated from data shared via GBIF and iDigBio, plus the best-practice georeferences produced in the large-scale collaborative georeference projects that were precursors to VertNet (MaNIS, HerpNet, and ORNIS). The API currently has one endpoint:

<https://localityservice.uc.r.appspot.com/api/bestgeoref>

which accepts POST requests formatted in JavaScript Object Notation (JSON). The basic structure required in the best georeference request is:

```
{'give_me':'best_georef', 'for_location': {}}
```

where the for_location must be filled with key:value pairs for the input fields to be used to find a match. The characteristics and treatment of input fields is the same as described in the Web App sections [Field Names](#) and [Field Name Modifications](#), above. Following is an example JSON request:

```
'{"give_me": "BEST_GEOREF", "for_location": {"ID": "1", "continent": "Europe", "country": "United Kingdom", "countrycode": "UK", "stateprovince": "England", "county": "Kent County", "locality": "Barnworth"}}'
```

One can test the API from a command line interface using curl, as follows:

```
curl -X POST -H "Content-Type: application/json" -d '{"give_me": "BEST_GEOREF",
"for_location": {"ID": "1", "continent": "Europe", "country": "United Kingdom", "countrycode": "UK",
"stateprovince": "England", "county": "Kent County", "locality": "Barnworth"}}'
https://localityservice.uc.r.appspot.com/api/bestgeoref
```

Georeference Matcher API Output

The BELS Georeference Matcher API returns results as JSON. A successful match will return a response with status "success" and the result of the matching process, which will include all of the fields in the "for_location" part of the request, plus fields added by the service (see the section [Georeference Matcher Field Descriptions](#)).

The result for the example request in the previous section, as of the writing of this manual, should be expected similar to the following:

```
{'Message': {'status': 'success', 'elapsed_time': '0.716s', 'result': {'ID': '1', 'continent': 'Europe', 'country': 'United Kingdom', 'countrycode': 'UK', 'stateprovince': 'England', 'county': 'Kent County', 'locality': 'Barnworth', 'bels_countrycode': 'GB', 'bels_match_string': 'gbenglandkentcountybarnworth', 'bels_decimallatitude': 51.179279, 'bels_decimallongitude': 0.630518, 'bels_geodeticdatum': 'epsg:4326', 'bels_coordinateuncertaintyinmeters': 21832, 'bels_georeferencedby': null, 'bels_georeferenceddate': '2019', 'bels_georeferenceprotocol': 'digital resource', 'bels_georeferencesources': 'GEOLocate', 'bels_georeferenceremarks': 'COGE georeferencing summer 2020: hforbes 7/9/2020 8:57:52 PM,', 'bels_georeference_score': 29, 'bels_georeference_source': 'iDigBio', 'bels_best_of_n_georeferences': 1, 'bels_match_type': 'match sans coords'}}}
```

During processing (see the section [BELS Georeference Matching Process](#), above), the country code "GB" was determined by looking up the value "UK" in the *countrycode* field in the input. You can see this choice reflected in the value of the output field *bels_countrycode*. The Georeference Matcher used that country code and constructed several strings by which to try to find a match. The string used to find the best successful match was "gbenglandkentcountybarnworth" as you can see in the *bels_match_string* field in the response. That string was constructed using the rules for "match sans coords", as can be seen in the value of the *bels_match_type*. The coordinates, datum and uncertainty were all derived from a georeference in the gazetteer aggregated out of iDigBio and having been originally generated in 2020 through the GEOLocate online collaborative georeferencing platform COGE. The original georeference was the only one used to generate the georeference returned in this response, as can be ascertained from the value "1" for the field *bels_best_of_n_georeferences*.

Georeference Matcher Field Descriptions

- **bels_match_country**: the value used to look up the ISO 3166-1 alpha-2 country code.
- **bels_interpreted_countrycode**: the value returned in the lookup of the match_country in the country lookup vocabulary (see <https://github.com/VertNet/bels/blob/main/bels/vocabularies/countrycode.txt>).
- **bels_matchwithcoords**: the string constructed to make a first attempt at finding a georeference for a Location with the same interpreted meaning. This field is based on the simplification of the content of all of the following input fields:
waterbody, islandGroup, island, stateProvince, county, municipality, locality, verbatimLocality, minimumElevationInMeters, maximumElevationInMeters, verbatimElevation, verticalDatum, minimumDepthInMeters, maximumDepthInMeters, verbatimDepth, verbatimCoordinates, verbatimLatitude, verbatimLongitude, decimalLatitude, decimalLongitude.
- **bels_matchverbatimcoords**: the string constructed to make a second attempt at finding a georeference for a Location with the same interpreted meaning. This field is based on the simplification of the content of all of the following input fields:
waterbody, islandGroup, island, stateProvince, county, municipality, locality, verbatimLocality, minimumElevationInMeters, maximumElevationInMeters, verbatimElevation, verticalDatum, minimumDepthInMeters, maximumDepthInMeters, verbatimDepth, verbatimCoordinates, verbatimLatitude, verbatimLongitude
- **bels_matchsanscoords**: the string constructed to make a second attempt at finding a georeference for a Location with the same interpreted meaning. This field is based on the simplification of the content of all of the following input fields:

waterbody, islandGroup, island, stateProvince, county, municipality, locality, verbatimLocality, minimumElevationInMeters, maximumElevationInMeters, verbatimElevation, verticalDatum, minimumDepthInMeters, maximumDepthInMeters, verbatimDepth

- **bels_decimallatitude:** the *decimalLatitude* of the best georeference returned by the service if a match is found.
- **bels_decimallongitude:** the *decimalLongitude* of the best georeference returned by the service if a match is found.
- **bels_geodeticdatum:** the *geodeticDatum* of the best georeference returned by the service if a match is found. The value of this field, if present, should always be "epsg:4326", corresponding to the WGS84 coordinate reference system.
- **bels_coordinateuncertaintyinmeters:** the *coordinateUncertaintyInMeters* of the best georeference returned by the service if a match is found and if that match contains a value for this field.
- **bels_georeferencedby:** the value of *georeferencedBy* of the best georeference returned by the service if a match is found and if that match contains a value for this field.
- **bels_georeferenceddate:** the value of *georeferencedDate* of the best georeference returned by the service if a match is found and if that match contains a value for this field.
- **bels_georeferenceprotocol:** the value of *georeferenceProtocol* of the best georeference returned by the service if a match is found and if that match contains a value for this field.
- **bels_georeferencesources:** the value of *georeferenceSources* for the best georeference returned by the service if a match is found and if that match contains a value for this field.
- **bels_georeferenceremarks:** the value of *georeferenceRemarks* for the best georeference returned by the service if a match is found and if that match contains a value for this field.
- **bels_georeference_score:** the score for the best georeference returned by the service based on the presence and completeness of georeference metadata. A more detailed description of the scoring system is available in the BELS GitHub repository at (<https://github.com/VertNet/bels>).

- **bels_georeference_source**: the data aggregation from which the best georeference returned by the service was derived (e.g., GBIF, iDigBio, VertNet).
- **bels_best_of_n_georeferences**: the number of original georeferences for the matching string from which the best georeference returned by the service was derived.
- **bels_match_type**: the method by which the best georeference returned by the service was determined. The possible values for this field in order of highest priority for being a best georeference are:
 - **original georeference** - the unmodified georeference provided in the input.
 - **match using coords** - the match was found using the string constructed with verbatim and decimal coordinates included.
 - **match using verbatim coords** - the match was found using the string constructed with verbatim coordinates included, but not with the decimal coordinates included.
 - **match sans coords** - the match was found using the string constructed without verbatim coordinates and without decimal coordinates included.
 - **original coordinates only** - no match for a best georeference was found using the three matching strings, so the original coordinates (without a fully specified georeference including both *geodeticDatum* and *coordinateUncertaintyInMeters*) was retained unmodified.
 - **match using coords - coords only** - none of the types of match above was successful, but the **match using coordinates** string was able to retrieve coordinates (without a fully specified georeference including both *geodeticDatum* and *coordinateUncertaintyInMeters*).
 - **match using verbatim coords - coords only** - none of the types of match above was successful, but the **match using verbatim coordinates** string was able to retrieve coordinates (without a fully specified georeference including both *geodeticDatum* and *coordinateUncertaintyInMeters*).
 - **match sans coords - coords only** - none of the types of match above was successful, but the **match sans coordinates** string was able to retrieve coordinates (without a fully specified georeference including both *geodeticDatum* and *coordinateUncertaintyInMeters*).