Vimmo User Manual

Contents **Page**

Panel Space: Please use this name space for exploration purposes. There are three endpoints associated with panel space as follows

1. GET /panels

Description

Returns **data** based on an input identifier (Panel_ID, Rcode, or HGNC_ID(s)). Providing multiple parameters will return a custom error message due to the complexity of handling these requests. It can also handle partial matches if the user sets Similar_Matches=true (defaults to False). However, this is still not implemented for HGNC ID. Please see below for the accepted format

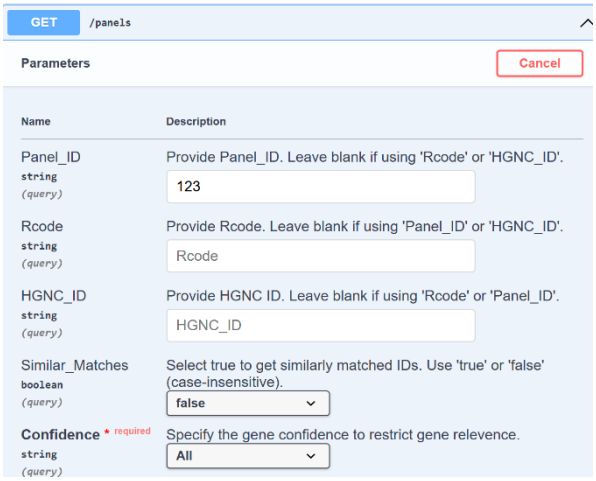
Accepted Query Parameters

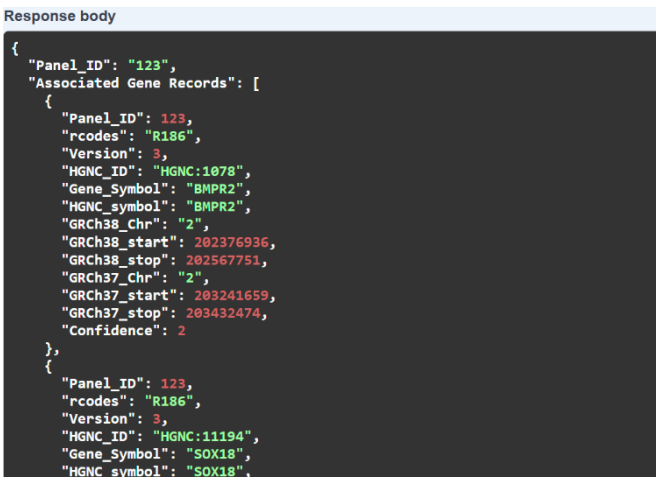
- **Panel_ID** (string/number, optional)
  Provide a numeric Panel ID (e.g., 123).
- **Rcode** (string, optional)
  Provide a code like R208.
- **HGNC_ID** (string, optional)
  Provide a single HGNC ID (e.g., HGNC:1100) or comma-separated multiple IDs (e.g., HGNC:1100,HGNC:2200,…).
- **Similar_Matches** (boolean, optional)
  If true, the endpoint attempts partial or "similar" matching (like LIKE queries in the DB). Defaults to false. This is not implemented when providing HGNC IDs and will raise an internal server error with 500 status code.
- **Confidence** (Choice (string), required)
  Based on the value, retrieves only genes that match the criteria, i.e, Green/Amber/Red or ALL (default value)

Return Format (JSON)

If one valid identifier is provided and matches the database records The API provides data based on the input parameter as follows:

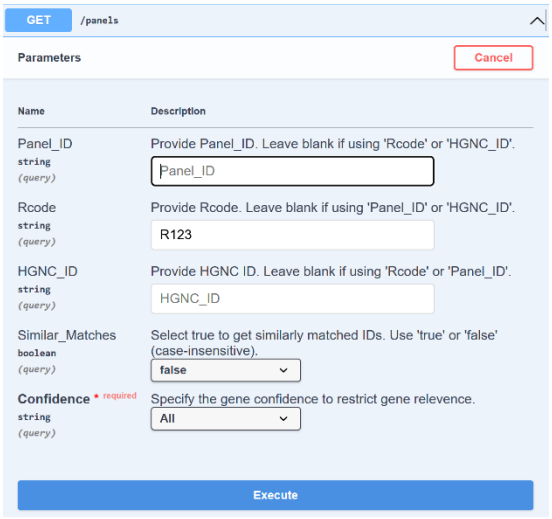
- If Panel ID:

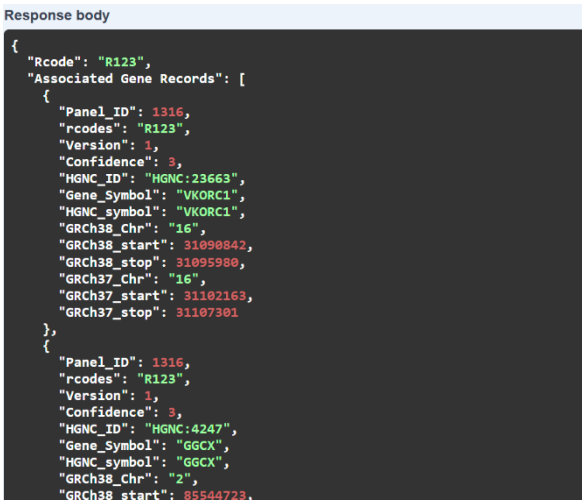




Returns a Json object with each element representing a gene information (HGNC ID, symbol, Grch 38/37 chromosome, start and stop), Any R code that utilises this panel ID and its associated version and the confidence of the gene within that R code.

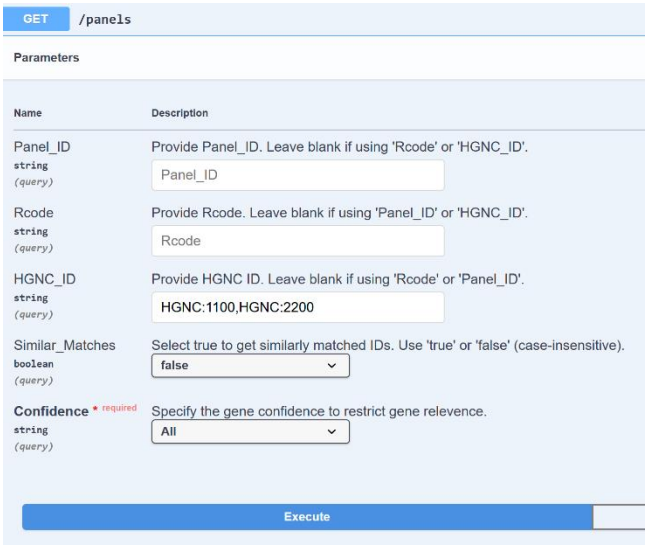
- If R Code provided:





Returns a Json object with each element representing a gene information (HGNC ID, symbol, Grch 38/37 chromosome, start and stop), all Panel IDs withing this rcode and its associated version and the confidence of the gene within that R code.

- If HGNC ID(s) provided:



Returns a Json object with all the associated panel IDs and R codes

- If **no matches** are found, you typically get:
- Returns a custom message to inform users of not matches found but with a 200 reponse

```
{
  "Panel_ID": "123456",
  "Message": "No matches found for this panel id with confidence : All."
}
```

- If you **pass multiple identifiers** at once (e.g., Panel_ID **and** Rcode), we raise a custom message with a 400 reposne as follows:

```
{
  "error": "Provide only one of 'Panel_ID', 'Rcode', or 'HGNC_ID', not multiple."
}
```

- If you **pass an invalid identifiers** with formatting issues, we raise custom error messages with 400 response to users to showcase the correc format

```
{
  "error": "Invalid input: 'Panel_ID' must be digits only (e.g., '123')."
}
```

---

2. GET /panels/download

Description
Generates and **returns a BED file** representing exome data from an **external** API call (e.g., the VariantValidator gene2transcripts_v2 endpoint). This means it queries an online service for each gene and then formats the result into a **BED** file.

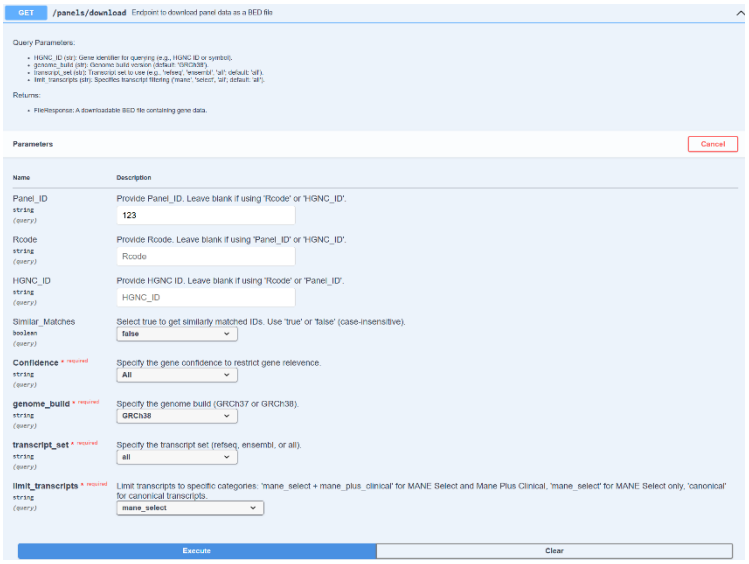
Accepted Query Parameters

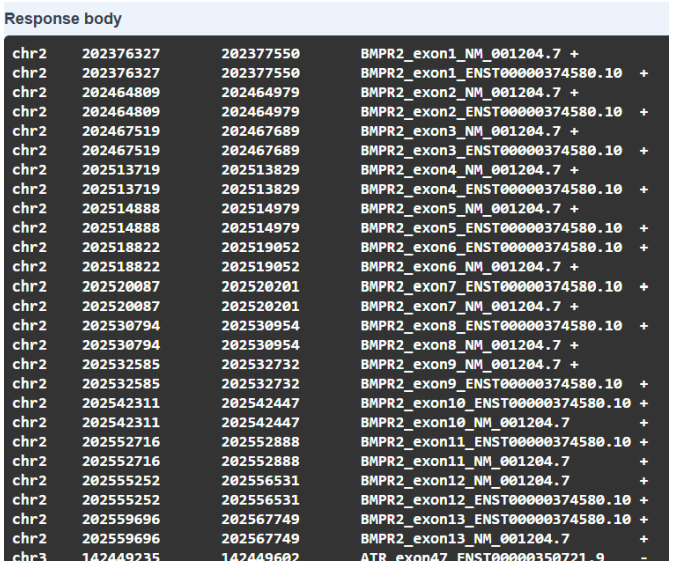
- **Panel_ID** or **Rcode** or **HGNC_ID**
    - Exactly one of these should be provided.
    - HGNC_ID can be a single ID or multiple comma-separated IDs.
- **Similar_Matches** (boolean, optional)
    - Same logic as /panels for partial matching. This does not raise an error and only matches the exact HGNC ID(s) when provided.
- **genome_build** (Choice(string), required)
    - Typically "GRCh38" or "GRCh37". Default is "GRCh38".

- **transcript_set** (Choice(string), required)
    - Usually "all", "refseq", or "ensembl".
    - Default is "all". Please see visti [] for more informaiton.
- **limit_transcripts** (string, optional)
    - e.g., "mane_select", "mane_plus_clinical", "all".
    - Default will be "mane_select".
    - "all" is equivalent to select option in variant validator api. Please visit [] for more information

Return Format

- **On success**:
    - A **BED file** (MIME type: text/plain) is returned as a downloadable attachment. Please copy the complete endpoint url and paste it into a new tab to download the file. This file should contain all the genes that make up a panel/rcode. IF HGNC ID(s) provided, it will retrieve only user-specified genes
    - Data consists of Chromosome, start, stop, <gene>_<exon>_<transcript id> , strandedness
    - The filename typically follows the user provided parameters.





- **On error**: returns a **JSON** with an "error" message and HTTP code **400** or **500**

---

3. GET /panels/download/local

Description
Similar to the **"download"** endpoint above, but uses a **local** BED records stored in our own database tables (bed37 or bed38). This does **not** call an external API. It relies on local copies of BED data. This endpoints should be used if contacting variant validator fails.

Accepted Query Parameters

- **Panel_ID** or **Rcode** or **HGNC_ID**
    - Must provide exactly one.
    - HGNC_ID (s) can be comma-separated.
- **Similar_Matches** (boolean, optional)
    - If true, partial matching for only Panel_ID or Rcode.
- **genome_build** (string, optional)
    - Either "GRCh37" or "GRCh38". If not specified, often defaults to "GRCh38".
- Possibly your code also requires:
    - **transcript_set** (e.g., "all")
    - **limit_transcripts** (e.g., "all")
      (Check your parser logic; these might be mandatory or defaulted.)

Return Format

- **On success**:
    - **A BED file** (MIME type text/plain) with local database data. However, the format is slightly different. As shown below:

- Return data consists of : chromosome, start, end , <gene>_<exon>, strandedness, transcript id (Gencode only), type (options[mane select (ms), mane plus clinical (mpc), canonical (can)], HGNC ID.

Response body

chr2	202376326	202377550	BMPR2_exon1	+	ENST00000374580.10	ms	HGNC:1078
chr2	202464808	202464979	BMPR2_exon2	+	ENST00000374580.10	ms	HGNC:1078
chr2	202467518	202467689	BMPR2_exon3	+	ENST00000374580.10	ms	HGNC:1078
chr2	202513718	202513829	BMPR2_exon4	+	ENST00000374580.10	ms	HGNC:1078
chr2	202514887	202514979	BMPR2_exon5	+	ENST00000374580.10	ms	HGNC:1078
chr2	202518821	202519052	BMPR2_exon6	+	ENST00000374580.10	ms	HGNC:1078
chr2	202520086	202520201	BMPR2_exon7	+	ENST00000374580.10	ms	HGNC:1078
chr2	202530793	202530954	BMPR2_exon8	+	ENST00000374580.10	ms	HGNC:1078
chr2	202532584	202532732	BMPR2_exon9	+	ENST00000374580.10	ms	HGNC:1078
chr2	202542310	202542447	BMPR2_exon10	+	ENST00000374580.10	ms	HGNC:1078
chr2	202552715	202552888	BMPR2_exon11	+	ENST00000374580.10	ms	HGNC:1078
chr2	202555251	202556531	BMPR2_exon12	+	ENST00000374580.10	ms	HGNC:1078
chr2	202559695	202567749	BMPR2_exon13	+	ENST00000374580.10	ms	HGNC:1078
chr3	142449234	142449602	ATR_exon47	-	ENST00000350721.9	ms	HGNC:882
chr3	142453127	142453233	ATR_exon46	-	ENST00000350721.9	ms	HGNC:882
chr3	142457603	142457755	ATR_exon45	-	ENST00000350721.9	ms	HGNC:882
chr3	142458957	142459111	ATR_exon44	-	ENST00000350721.9	ms	HGNC:882
chr3	142459226	142459383	ATR_exon43	-	ENST00000350721.9	ms	HGNC:882
chr3	142461939	142462090	ATR_exon42	-	ENST00000350721.9	ms	HGNC:882
chr3	142465096	142465240	ATR_exon41	-	ENST00000350721.9	ms	HGNC:882
chr3	142466323	142466533	ATR_exon40	-	ENST00000350721.9	ms	HGNC:882
chr3	142467933	142468068	ATR_exon39	-	ENST00000350721.9	ms	HGNC:882
chr3	142469336	142469569	ATR_exon38	-	ENST00000350721.9	ms	HGNC:882
chr3	142470085	142470183	ATR_exon37	-	ENST00000350721.9	ms	HGNC:882
chr3	142485139	142485282	ATR_exon36	-	ENST00000350721.9	ms	HGNC:882
chr3	142493131	142493311	ATR_exon35	-	ENST00000350721.9	ms	HGNC:882
chr3	142496360	142496520	ATR_exon34	-	ENST00000350721.9	ms	HGNC:882

Note: Type was used to simplify database creation of all bed records. GTF was used to extract all the necessary genes information. However, the complexity of having many transcript records resulted in having multiple records for some exons, which can be confusing. To overcome this more simply. The records were filtered based on priority. If ms is found for a given exon, the database will only have these records and ignores mpc and can. If ms not found, the database will then try to get any mpc records if exist and will ignore canonical. Finally If neither records were found then the database will only retrieve any canonical records. This prevents creating duplicate records which can interfere with sorting downstream. It is also important to note that ms and mpc can also be considered canonical.

- if no data found or missing parameters returns a custom message.

Patient Space: Please use to query existing patient data mimicking a LIMS system. This name space has three endpoints that is able to retrieve all records on patients. Create an automated bed file either via using external services or locally.

4. GET /patient

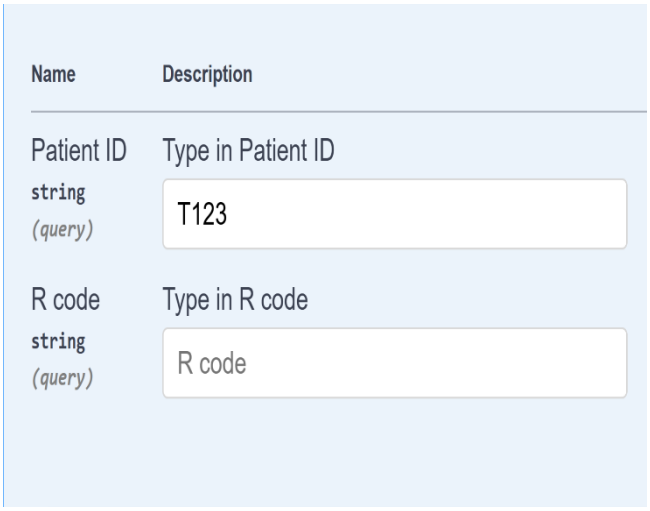
Description
Returns **patient test records**. If only a Patient ID is given, it lists **all R codes** (plus versions/dates) that patient has. If only an R code is given, it lists **all patients** who have that code. If both parameters are input and a change in version is detected between the patients last R code and the current version, a panel comparison is returned to the user.

Accepted Query Parameters

- **Patient ID** (string, optional)
    - e.g., T123.
- **R code** (string, optional)
    - e.g., R208.

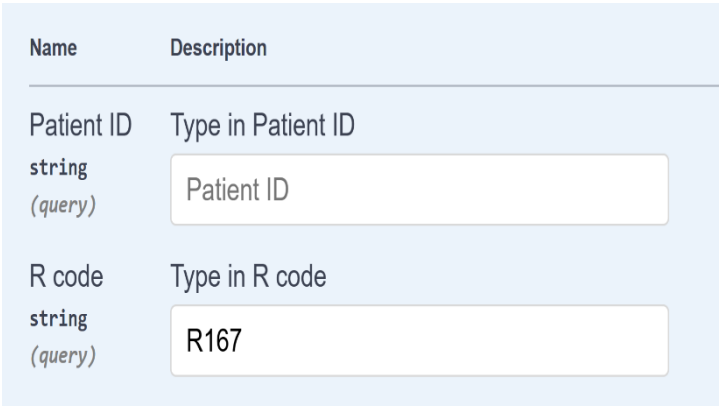
Return Format (JSON)

A) Only Patient ID Provided



- Each key in "patient records" can be an index (0, 1, 2...) whose value is a date → [Rcode, version].

B) Only R code Provided



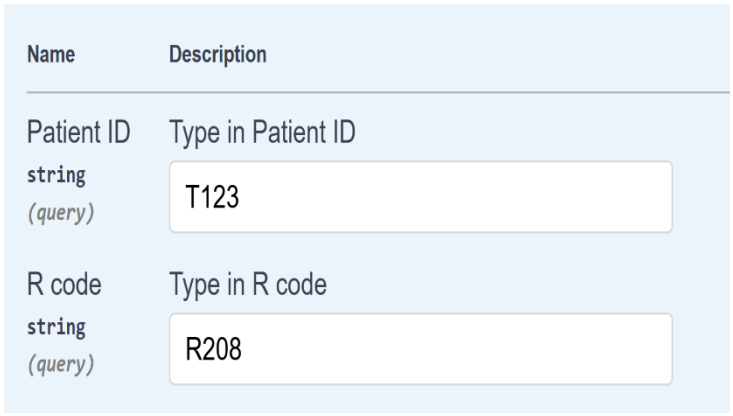

```
{
  "R code": "R167",
  "Records": {
    "0": {
      "2022-5-16": [
        "T456",
        1.5
      ]
    },
    "1": {
      "2024-5-17": [
        "T456",
        1.5
      ]
    }
  }
}
```


- Each key in "Records" is an index (0, 1, 2...) whose value is a date → [Patient_ID, version].

C) Both Provided

(Your code might do version checks or simply return a "no version change" message if everything is up to date.)

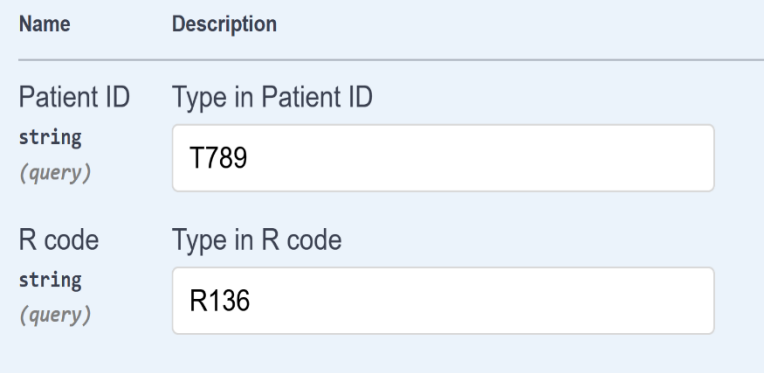
Example of no change in version detected.




```
{
  "disclaimer": "Panel comparison up to date",
  "status": "No version change since patient: T123 last had R208",
  "Version": "2.5",
  "current_panel": {
    "HGNC:795": 3,
    "HGNC:1100": 3,
    "HGNC:1101": 3,
    "HGNC:16627": 3,
    "HGNC:26144": 3,
    "HGNC:9820": 3,
    "HGNC:9823": 3
  },
  "Tip": "For more information on R208 v2.5, please use the panels space."
}
```

Here, the version of R208 hasn't changed since the patient last had an R208.

Alternatively, in the event the panel version has changed, the contents of the panels will be compared and displayed to the user.




```
{
  "disclaimer": "Panel comparison up to date",
  "status": "Version changed since last T789 had R136",
  "Version": "3.0",
  "Genes added": {
    "HGNC:485": 2,
    "HGNC:24097": 2,
    "HGNC:3446": 3,
    "HGNC:13604": 1,
    "HGNC:10260": 2,
    "HGNC:11809": 2
  },
  "Genes removed": {},
  "Confidence changes (old ver -> new ver)": {
    "HGNC:20626": [
      3,
      1
    ]
  }
}
```


Breakdown of the output

"disclaimer" – This value states whether the database was updated at the time of the query. In the above example, the panelapp client could be contacted and therefore the Vimmo database is up to date. If not updated the disclaimer will display: 'Database update failed – using most recent db version (e.g 2.5) for panel comparison'

"Status" – Here, the presence of a version change is confirmed or denied. Here the version

"Version" - The current/most recent version of the panel in the database

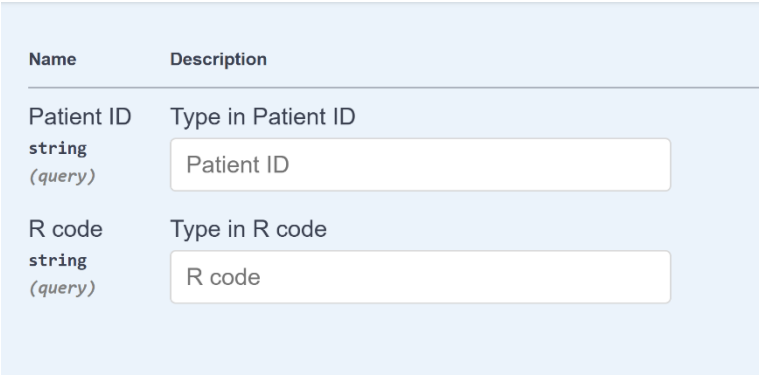
"Genes added" – The genes present in the new version of the panel that were not present when the patient last received the panel. On each line is the HGNC ID and confidence score of the gene.

"Genes removed" – The genes not present in the newer version of the panel.

"Confidence changes" – Genes the have remained of the panel, however there panelapp confidence has changed (1= Red, 2= Amber, 3= Green).

D) Edge Cases

- **If neither Patient ID nor R code is given**: Some code returns a **200** with:




```
{
  "error": "At least one of 'Panel_ID' or 'Rcode' must be provided. None"
}
```

or you might choose to return a **400** if you want to enforce that at least one param is required.

---

5. GET /patient/bed

- **Purpose**
    - Creates a BED file for a given patient and optionally a specific R code & version, calling the VariantValidator API. Note that, it only returns a bed file if the patient has a single instance of record or else asks user to be more specific based on scenarios.
    - Similar to /panels/download but specifically for a patient's previously recorded R code and version.
- **Query Parameters (Required)**
    - Patient ID (string)
    - genome_build = GRCh37 or GRCh38
    - transcript_set = refseq, ensembl, or all
    - limit_transcripts = 'mane_select + mane_plus_clinical', 'mane_select', 'canonical'
- **Query Parameters (Optional)**
    - R code (string)
    - version (string, e.g. 2.5)
    - Padding (int, default=0, max=250) for adding bases around intervals.
- **Logic Flow:**
        - If Patient id provided and R code/version missing, tries to auto-detect all records from the database records records and returns a json object with

multiple records (if present) or a bed file if only a single associated record is identified. If multiple or none → returns a custom message to guide user to provide more parameters.

- If R code provided and version missing but multiple versions records were associated with the patient → returns a message to specify the version the user would like to create the bed for.
- If both provided → checks that (patient_id, r_code, version) actually exists in patient_data.
- Checks whether version is out of date vs. DB's newest version. If older → uses archived genes from panel_genes_archive; if current → uses panel_genes to create a list of genes.
- Calls VariantValidator with the final gene list.
- Returns a .bed file named: "<patient_id>_<r_code>_<genome_build>_<limit_transcripts>.bed".

- **Response Examples**
  - Successful: HTTP 200, text/plain with BED data. Patient ID: T123 with a single R208 records**:**

Response body

```
chr11   108223067       108223186       ATM_exon1_NM_000051.4   +
chr11   108227595       108227696       ATM_exon2_NM_000051.4   +
chr11   108227776       108227888       ATM_exon3_NM_000051.4   +
chr11   108229178       108229323       ATM_exon4_NM_000051.4   +
chr11   108235670       108235834       ATM_exon5_NM_000051.4   +
chr11   108243953       108244118       ATM_exon6_NM_000051.4   +
chr11   108244788       108245026       ATM_exon7_NM_000051.4   +
chr11   108246964       108247127       ATM_exon8_NM_000051.4   +
chr11   108248933       108249102       ATM_exon9_NM_000051.4   +
chr11   108250701       108251072       ATM_exon10_NM_000051.4  +
chr11   108251837       108252031       ATM_exon11_NM_000051.4  +
chr11   108252817       108252912       ATM_exon12_NM_000051.4  +
chr11   108253814       108254039       ATM_exon13_NM_000051.4  +
chr11   108256215       108256340       ATM_exon14_NM_000051.4  +
chr11   108257481       108257606       ATM_exon15_NM_000051.4  +
chr11   108258986       108259075       ATM_exon16_NM_000051.4  +
chr11   108267171       108267342       ATM_exon17_NM_000051.4  +
chr11   108268410       108268609       ATM_exon18_NM_000051.4  +
chr11   108271064       108271146       ATM_exon19_NM_000051.4  +
chr11   108271251       108271406       ATM_exon20_NM_000051.4  +
chr11   108272532       108272607       ATM_exon21_NM_000051.4  +
chr11   108272722       108272852       ATM_exon22_NM_000051.4  +
chr11   108279491       108279608       ATM_exon23_NM_000051.4  +
chr11   108280995       108281168       ATM_exon24_NM_000051.4  +
chr11   108282710       108282879       ATM_exon25_NM_000051.4  +
chr11   108284227       108284473       ATM_exon26_NM_000051.4  +
chr11   108287600       108287715       ATM_exon27_NM_000051.4  +
```

  - Multiple records found when no R code was provided: Patient ID:

Response body


```
{
  "MESSAGE": "Multiple records found",
  "Patient ID": "T456",
  "patient records": {
    "0": {
      "2022-5-16": [
        "R167",
        1.5
      ]
    },
    "1": {
      "2024-5-17": [
        "R167",
        1.5
      ]
    },
    "2": {
      "2024-5-17": [
        "R100",
        5
      ]
    }
  },
  "Tip": "Please select a panel and version"
}
```


- If patient ID does not match our records the json object consists of following message:

Response body


```
"Please use the update space as no records were found for T999"
```


- No BED failed to generate for some reason will return :

{

"error": "No BED data could be generated from the provided gene query.",

"Tip": "Please use local bed endpoint…"

}

- **Please report this by raising an issue as this needs to be investigated**

- External API error custom API error with the following message:

```
{
 "error": "VarValAPIError: ..."
}
```

  - (HTTP 400 or 500 depending on the cause.)

---

6. GET /patient/local_bed (Local)

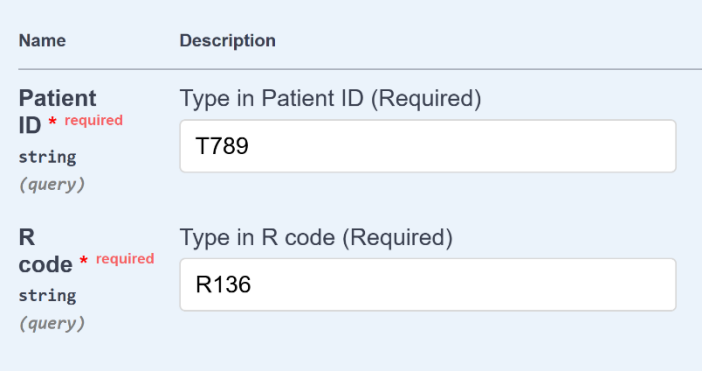
- **Purpose**
  - Also produces a BED file for a patient's R code & version, but uses local DB tables (bed37/bed38) for coordinates, not VariantValidator.
  - Parallel to /patient/bed but fully local. Please refer to section 5 for more comprehensive details.
- **Key Points**
  - Logic Flow: same as /patient/bed. Please refer to section 5 for more comprehensive details.
  - Infers (R code, version) from patient_data if not provided, or uses user's specified R code & version.
  - If multiple records or none, returns a message.
  - With the final gene list, queries bed37 or bed38 (depending on genome_build) and applies any Padding.
- **Query Parameters**
  - Patient ID (required),
  - Optional R code, version,
  - genome_build (GRCh37 or GRCh38),
  - Padding (int, default=0, max=250).
- **Output**
  - A local .bed file named "<patient_id>_<r_code>_<genome_build>_Gencode.bed".
  - 400 if no data or if something is missing

UpdatePatientRecords : Please use this endpoint to update the Vimmo database with new patient test records.

7. GET UpdatePatientRecords

- **Purpose**
  - **Update the Vimmo database with a patient testing record**
  - *** This feature assumes the day of entry MATCHES into the DB is the day of patient testing ***
- **Query Parameters**
  - **R_Code (required)**
  - **Patient_ID (required)**
- **Flow:**
    - **Checks the panel version is up to date. (Updates if necessary)**
    - **Checks if the patient record already exists (record matching Patient ID, R code and Version)**
    - **In absence of pre-existing record, add record of patient test into the database.**
- **Response**
  - **Str – Explanatory message of new entries details.**
- **Response examples**

Successful record entry:




```
"Record added to database: Patient_id: T789, Rcode: R136, version: 2.2, date: 2025-01-07"
```

Unsuccessful record entry:

R codes must exist within panel app to be accepted, otherwise the following error will be raised.

```
{
  "error": "Rcode: R200 not within our records - please select a valid rcode"
}
```

Patient ID's must be alphanumeric. Inputs failing this constraint will raise the following error.

```
{
  "error": "Invalid format for 'Patient_ID': Must be alphanumeric."
}
```

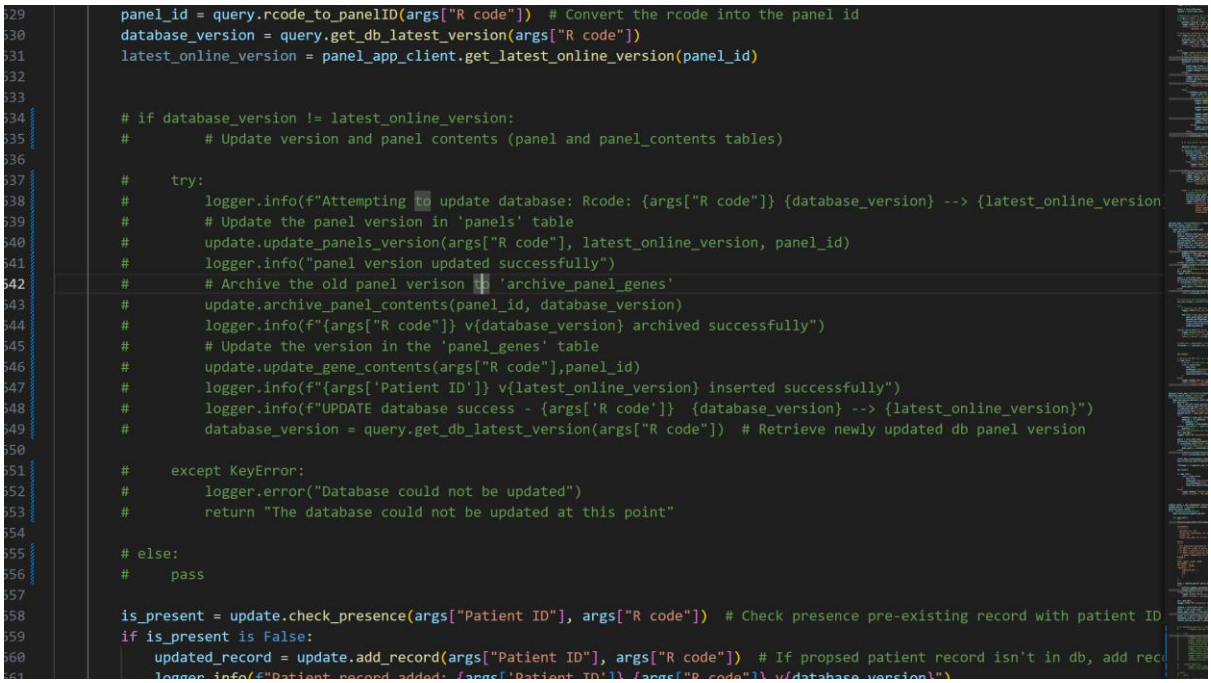
Downgrade space: WARNING !!!!!!!! This is for development purposes only to test out the auto update functionality of the database. Use it at your own discretion as this is not fully tested. Using this inappropriately can lead to unexpected database alteration and break application functionality. Use within a live environment could also negatively impace patient care.

8. GET /Downgraded (Local)

- **Purpose**
  - **Functions to revert a panel version based on user provided input queries.**
- **Query Parameters**
  - **R_Code (required)**
  - **version (required)**
- **Flow:**
    - **Checks the DB's current version for that R_Code.**
    - **If it matches the user's version, returns "Requested version matches current database version."**
    - **Otherwise, attempts to fetch that older version's records from PanelApp rewrite panel_genes.**
- **Response**
  - **JSON describing the changes (added, removed, updated) or an error if no records are found or the transaction fails.**

Note: Since the UpdateSpace and Patient Space auto update the database every time the user requests an endpoint. Certain steps need to be taken first:

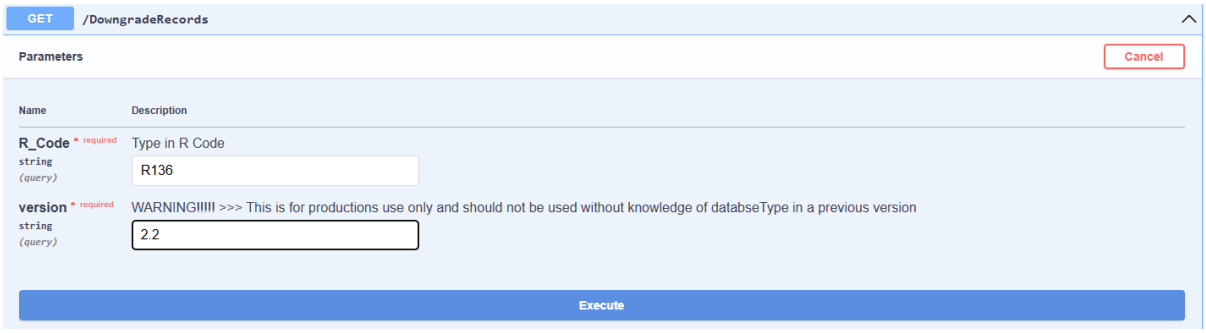
Step 1: Disable the auto update functionaly by commenting out the code in the endpoint.py source code (lines634 – lines 656):



```
        panel_id = query.rcode_to_panelID(args["R code"])  # Convert the rcode into the panel id
        database_version = query.get_db_latest_version(args["R code"])
        latest_online_version = panel_app_client.get_latest_online_version(panel_id)


        # if database_version != latest_online_version:
        #         # Update version and panel contents (panel and panel_contents tables)

        #     try:
        #         logger.info(f"Attempting to update database: Rcode: {args["R code"]} {database_version} --> {latest_online_version
        #         # Update the panel version in 'panels' table
        #         update.update_panels_version(args["R code"], latest_online_version, panel_id)
        #         logger.info("panel version updated successfully")
        #         # Archive the old panel verison to 'archive_panel_genes'
        #         update.archive_panel_contents(panel_id, database_version)
        #         logger.info(f"{args["R code"]} v{database_version} archived successfully")
        #         # Update the version in the 'panel_genes' table
        #         update.update_gene_contents(args["R code"],panel_id)
        #         logger.info(f"{args['Patient ID']} v{latest_online_version} inserted successfully")
        #         logger.info(f"UPDATE database success - {args['R code']}  {database_version} --> {latest_online_version}")
        #         database_version = query.get_db_latest_version(args["R code"])  # Retrieve newly updated db panel version

        #     except KeyError:
        #         logger.error("Database could not be updated")
        #         return "The database could not be updated at this point"

        # else:
        #     pass

        is_present = update.check_presence(args["Patient ID"], args["R code"])  # Check presence pre-existing record with patient ID
        if is_present is False:
            updated_record = update.add_record(args["Patient ID"], args["R code"])  # If propsed patient record isn't in db, add rec
            logger.info(f"Patient record added: {args['Patient ID']} {args["R code"]} v{database_version}")
```

save the code and restart the application to reflect these changes.

Step 2: use the Downgrade Space to downgrade a specific R code to an existing version e.g, the current database version for R136 on our application is 3 and the previous version being (2.2), so provide the parameters R136 and 2.2:



Give the following response:

Response body


```
{
  "panel_id": 65,
  "rcode": "R136",
  "previous_version": 3,
  "new_version": "2.2",
  "changes": {
    "added": [],
    "removed": [
      "HGNC:485",
      "HGNC:24097",
      "HGNC:10260",
      "HGNC:11809",
      "HGNC:13604"
    ],
    "confidence_changed": [
      {
        "hgnc_id": "HGNC:20626",
        "old_confidence": 1,
        "new_confidence": 3
      }
    ]
  }
}
```


Panel_ID	rcodes	Version
25	null	6.7
31	R145	2.2
49	R131	4.16
53	R101	3.0
55	null	2.0
59	null	4.0
65	R136	3.0

Panel_ID	rcodes	Version
25	null	6.7
31	R145	2.2
49	R131	4.16
53	R101	3.0
55	null	2.0
59	null	4.0
65	R136	2.2

Step 3: Update the database with a patient record with the auto update functionality disabled to add a patient record with the downgraded version. The response should reflect the version added:

Name Description

Patient ID * required
string
(query)
Type in Patient ID (Required)
T123

R code * required
string
(query)
Type in R code (Required)
R136

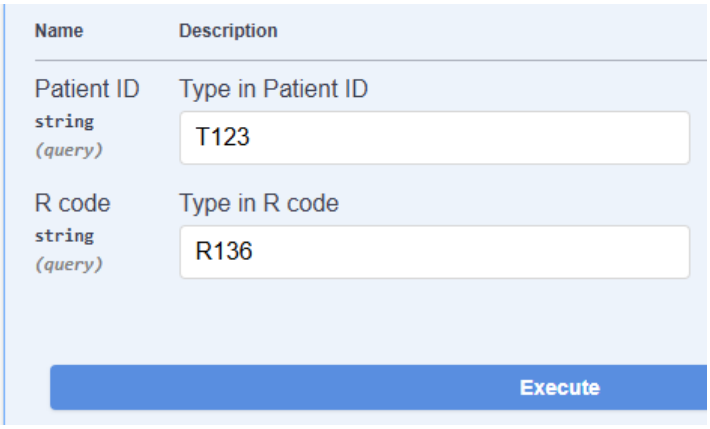
Execute

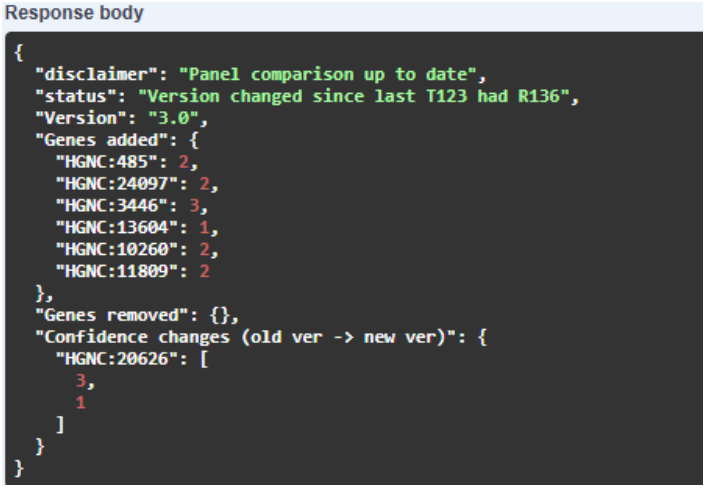
Response body

```
"Record added to database: Patient_id: T123, Rcode: R136, version: 2.2, date: 2025-01-07"
```

Patient...	Panel_ID	Rcode	Version	Date
T123	635	R208	2.5	2023-12...
T456	557	R167	1.5	2022-5-16
T456	557	R167	1.5	2024-5-17
T456	168	R100	5.0	2024-5-17
T789	65	R136	2.2	2025-01...

Step4: Now testing the patient in the Patient Space should autoupdate to the latest version and inform users of a change in panel:





Step 5: Request a BED for this patient by typing in the ID and R code with version>> This should retrieve data from the archived panel table.

If a process fails for various reason, please reset the database and raise an issue to report the issue.

To reset change directory to <your_file_path>/SoftwareDevelopmentVIMMO/database_prework/created/

RUN $python create_newdb.py

Once database is reset change into to root directory (<your_file_path>/SoftwareDevelopmentVIMMO/) and run the app as normal.