Patent Analysis For Norwegian applications

- File Structure & Protocol Project Directory Structure

Team-analyse/

```
│
├── main.py                          # MAIN ORCHESTRATOR (entry point)
├── config.py                        # Configuration settings (API keys, paths, log levels)
├── validators.py                    # Input validation functions
│
├── get_family_ids.py                # Step 1: Fetch patent family IDs from database
├── get_main_table.py                # Step 2: Fetch application data + priority authority
├── get_classes.py                   # Step 3: Fetch IPC/CPC classification data
├── get_applicants_inventors.py      # Step 4: Fetch applicants & inventors data
│
├── data_analysis_applicants_inventors.py   # Data processing for applicants/inventors
│
├── ipc_technology_eng.xlsx          # LOOKUP FILE: IPC codes → Sector & Field mapping
│
└── DataTables_[COUNTRY]_[YEAR1]_[YEAR2]/   # OUTPUT DIRECTORY (created per analysis)
    ├── family_ids.csv               # Step 1 output: List of patent family IDs
    ├── main_table.csv               # Step 2 output: All applications & basic data
    ├── main_table_priority_classes.csv     # Step 2 + Step 3 merged: Applications + Classifications
    ├── temp_main_table_priority_classes.csv  # Temporary file for aggregation
    ├── main_table_agg.csv           # **FINAL OUTPUT**: Aggregated by family with all enrichments
    └── [analysis_results_final.csv]        # (Alternative final output with additional metrics)
```

Data Processing Protocol - Step by Step

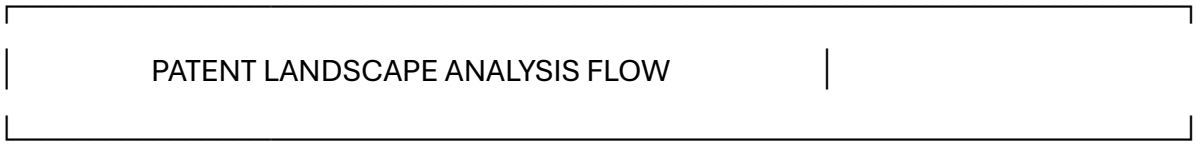
INPUT

Country Code (e.g., "NO")

Year Range (e.g., 2020-2022)

Optional: Limit number of families

PROCESSING PIPELINE



STEP 1: FETCH FAMILY IDs

├─ Query patent database by country & year

├─ Get unique docdb_family_id values

└─ Output: family_ids.csv (ID list)

↓

STEP 2: FETCH MAIN APPLICATION DATA

├─ For each family ID, get all applications

├─ Collect: appln_auth, filing_date, applicant count, inventor count

├─ Merge in: priority_auth (from priority chain analysis)

├─ Apply fallback logic for empty priority_auth:

│ └─ If appln_auth == auth_family (single authority), use as priority_auth

└─ Output: main_table.csv

↓

STEP 3: FETCH CLASSIFICATION DATA

├─ Get IPC (International Patent Classification) for each family

├─ Get CPC (Cooperative Patent Classification) for each family

├─ Output: classification data merged into main table

↓

STEP 4: FETCH APPLICANTS & INVENTORS DATA

├─ Get names & locations of all applicants

├─ Get names & locations of all inventors

├─ Count by country per family

└─ Separate CSV: applicants_inventors_by_country.csv (for team analysis in Excel)

↓

STEP 5: AGGREGATE & ENRICH

├– Group all data by docdb_family_id

├– Create auth_family: comma-separated list of all authorities in family

├– Load IPC mapping file (ipc_technology_eng.xlsx)

├– Match main_ipc_group to get:

│ ├– sector (e.g., "Chemistry", "Mechanical engineering")

│ └– field (e.g., "Pharmaceuticals", "Handling")

├– Column ordering:

│ ├– KEY: docdb_family_id, application_number, appln_auth, auth_family, priority_auth

│ ├– DATA: (other columns like filing_year, family_size, etc.)

│ ├– CLASSES: (main_ipc_group, cpc_classes)

│ └– ENRICHMENT: sector, field

└– Output: main_table_agg.csv **[FINAL DELIVERABLE]**

↓

OUTPUT: main_table_agg.csv

├─ One row per family

├─ All metadata enriched

├─ Sector & Field classified

└─ Ready for analysis

File Descriptions

INPUT FILES

ipc_technology_eng.xlsx (Lookup Reference)

- **Purpose**: Maps IPC codes to business sectors and fields
- **Columns**:
    - IPC_code: IPC classification codes (e.g., A61K 31/%, B65G%)
    - Sector_en: Technology sector (e.g., "Chemistry", "Mechanical engineering")
    - Field_en: Technology field (e.g., "Pharmaceuticals", "Handling")
- **Format**: Uses % as wildcard for pattern matching
- **Normalization**: Spaces and / characters are stripped for matching

---

OUTPUT FILES

main_table_agg.csv [PRIMARY OUTPUT]

- **Purpose**: Single comprehensive dataset ready for analysis
- **Granularity**: One row per patent family
- **Key Columns**:
- docdb_family_id - Unique patent family identifierapplication_number - First application number in familyappln_auth - Application authorityauth_family - All authorities in family (comma-separated)priority_auth - Earliest/priority authority[other data columns] - Filing year, family size, granted status, etc.main_ipc_group - IPC classificationcpc_classes - CPC classificationssector - Business sector (from IPC mapping)field - Technology field (from IPC mapping)

applicants_inventors_by_country.csv [FOR EXCEL ANALYSIS]

- **Purpose**: Raw data for analysis person to combine counts & ratios as needed
- **Granularity**: One row per (family, country) combination
- **Columns**:
- docdb_family_id - Patent family IDperson_ctry_code - Country code (NO, US, DE, etc.)applicant_count - Number of applicants from this countryinventor_count - Number of inventors from this country
- **Advantage**: Analysis person can calculate ratios in Excel using their own logic

---

Data Flow Diagram

Key Processing Logic

1. Priority Authority Fallback

IF priority_auth is empty:

IF appln_auth == auth_family (same single authority):

priority_auth = appln_auth

ELSE:

priority_auth = "Unknown"

2. Auth Family Aggregation

Group all rows by docdb_family_id

auth_family = sorted comma-separated unique appln_auth values

Example: "CA, CL, DK, EP, JP, US, WO"

3. IPC to Sector/Field Mapping

Normalize both IPC code and pattern:

- Remove all spaces
- Remove everything after '%' or '/'
- Convert to uppercase

Match normalized versions:

"A61K 31" → "A61K31"

Pattern "A61K 31/%" → "A61K31"

→ MATCH ✓ → Get sector & field

4. Column Organization

[Key Identifiers] → [Data Columns] → [Classifications] → [Enrichment]

docdb_family_id, application_number, appln_auth, auth_family, priority_auth,

[appln_filing_year, docdb_family_size, granted, ...],

[main_ipc_group, cpc_classes],

[sector, field]

Configuration & Setup

config.py

Contains:

- Database connection parameters
- API credentials
- File paths
- Log level settings
- Output directory base path

validators.py

Validates:

- Country code format (2-letter ISO)
- Year range validity
- Input parameter checks

Execution

Run Full Analysis

python main.py

Parameters (in main.py at bottom)

COUNTRY = "NO" # Country code

START_YEAR = 2020 # Start year

END_YEAR = 2020 # End year

RANGE_LIMIT = None # Optional: limit families to process

Output Location

DataTables_NO_2020_2020/

├── family_ids.csv

├── main_table.csv

├── main_table_priority_classes.csv

├── main_table_agg.csv ← **USE THIS FOR ANALYSIS**

└── applicants_inventors_by_country.csv

Quality Checks

- ✓ No missing sector/field (mapped from IPC)
- ✓ No empty priority_auth (filled with fallback logic)
- ✓ auth_family shows all authorities in each family
- ✓ Data aggregated correctly by family ID
- ✓ Columns organized logically for analysis