



УДК 519.688:581.9:581.5

<https://doi.org/10.53904/1682-2374/2024-26/14>

Н.О. Скобель^{1,2}, В.В. Шаповал³

¹*Херсонський державний університет*

вул. Шевченка, 14, м. Івано-Франківськ, 77333 Україна

²*Варшавський університет*

вул. Жвїркі і Вігури, 101, м. Варшава, 02-089 Польща

³*Біосферний заповідник "Асканія-Нова" імені Ф.Е. Фальц-Фейна НААН*

вул. Метрологічна, 12, м. Київ, 03143 Україна

¹e-mail: skobel@gmail.com

³e-mail: shapoval_botany@ukr.net

¹<https://orcid.org/0000-0003-4354-2881>

³<https://orcid.org/0000-0003-0443-663X>

ПЕРЕТВОРЕННЯ ГЕОБОТАНІЧНИХ ОПИСІВ У LONG DATA FORMAT ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ PYTHON

Біорізноманіття, GBIF, Darwin Core, геоботанічні дані, програмування, середовище розробки

ПЕРЕТВОРЕННЯ ГЕОБОТАНІЧНИХ ОПИСІВ У LONG DATA FORMAT ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ PYTHON. Н.О. Скобель, В.В. Шаповал. –

У статті розглянуто методичні засади з перетворення геоботанічних описів у Long Data Format відповідно до стандартів Darwin Core за допомогою мови програмування Python у середовищі PyCharm. Вперше в Україні розроблено спеціалізований скрипт на Python, що реалізує дані методологічні підходи. Подано детальні інструкції встановлення програмного забезпечення, необхідних бібліотек та підготовки даних до подальшої трансформації. Зазначено переваги та недоліки використання Python у порівнянні з мовою програмування R у цьому контексті. Висвітлено основні переваги при автоматизації процесів: швидкість виконання коду, масштабованість, повторюваність у використанні, мінімізація помилки та ефективність управління даними на противагу ручному методу переведення даних. Надано практичні рекомендації для наукових працівників, які займаються обробкою геоботанічної інформації. Розробка скриптів для роботи з геоботанічними наборами даних та їх впровадження сприятиме збереженню та інтеграції результатів досліджень у відкритий простір, зокрема при публікації у Global Biodiversity Information Facility (GBIF).

TRANSFORMATION OF RELEVÉS INTO LONG DATA FORMAT USING PYTHON PROGRAMMING LANGUAGE. N.O. Skobel, V.V. Shapoval. –

This article presents methodological principles for transforming relevés into Long Data Format following Darwin Core standards using Python programming language in the development environment PyCharm. For the first time for Ukraine, a specialized Python script has been developed that implements these methodological approaches. The detailed instructions for installing the software and the necessary libraries, as well as data preparation, are provided. We also outline the advantages and disadvantages of using Python compared to the R programming language in this context. The main advantages of automating processes are highlighted: code execution speed, scalability, repeatability, error minimization and data management efficiency compared to the manual method of data conversion. The practical recommendations for researchers involved in relevés datasets processing are provided. The development of scripts for relevés datasets and their implementation will facilitate the preservation and integration of research results in Open Access, including publishing through the Global Biodiversity Information Facility (GBIF).

Оцифрування геоботанічних даних відкриває нові можливості для аналізу, інтеграції, обміну та забезпечення їх збереження у відкритому доступі, що актуалізується у контексті втрат біорізноманіття та ймовірності втрати даних про біорізноманіття під час повномасштабного вторгнення росії в Україну. У зоні найбільшого ризику перебувають території та об'єкти природно-заповідного фонду на лінії бойового зіткнення і тимчасово окуповані, до яких належить Біосферний заповідник "Асканія-Нова".

Автоматизація операцій з перетворення, очищення та стандартизації геоботанічних описів значно прискорює процес обробки даних та забезпечує високу точність набору, мінімізуючи ризик помилок, спричинених людським фактором. Водночас автоматизовані інструменти є масштабованими, дають змогу адаптуватися до великих обсягів даних і повторно використовувати розроблені скрипти.

Геоботанічні описи містять інформацію про поширення видів у часі та просторі з супутніми характеристиками (площа та локалізація, режим використання описових ділянок, загальне та осібне проєктивне покриття, метод реєстрації, фенофази рослин тощо). Подібні дані відповідають вимогам саме Sampling-event (набір даних про подію) у Global Biodiversity Information Facility (GBIF) з рекомендованими розширеннями Occurrence та Relevé та ін. Для публікації набору даних у GBIF останні мають бути представлені у Long Data Format (або "довгий формат даних") – це спосіб організації даних у табличному вигляді, де кожен рядок являє собою одну спостережувану одиницю (знахідка в геоботанічному описі), який є стандартом для GBIF (Wieczorek et al., 2012). На противагу, одним з основних способів збереження геоботанічних описів в Україні є Wide Data Format (або "широкий формат даних"). У Wide Data Format змінні розподілені у колонках, тобто, у кожній колонці міститься інформація про знахідку виду в межах геоботанічного опису з оцінкою його проєктивного покриття. Геоботанічні описи в Wide Data Format – безперечно важливе джерело інформації про рослинний покрив, що потребує адаптації до сучасних вимог відкритої науки та адаптації дослідницьких даних до Long Data Format. Ми акцентуємо увагу на важливості збереження даних в обох форматах для альтернативних узагальнень та інтерпретацій. Зважаючи на практичну потребу автоматизації цих процесів та актуальність відповідних методичних ініціатив, у поданій статті вперше представлено спеціалізований скрипт на Python для автоматизації перетворення геоботанічних описів у Long Data Format відповідно до стандартів Darwin Core (Wieczorek et al., 2012). Необхідно наголосити, що при переформатуванні подібного набору за допомогою програми Excel зростає ймовірність похибки "випадкового зайвого протягування рядків" між різними геоботанічними описами, продовження автоматичної нумерації для координат, випадкове видалення / заміщення / дублювання опису та ін. Крім того, з нашого попереднього досвіду для такого великого об'єму даних переформатування може зайняти принаймні 100 годин часу. Натомість, код дозволяє зробити це впродовж лічених секунд через створення аркуша з розширенням Occurrence.

В інформаційних технологіях активно використовують 2 мови програмування для автоматизації процесів та обробки даних – R та Python (R Core Team, 2024; Python Software Foundation, 2024), але кожна з них має свої особливості (McKinney, 2018). R створено спеціально для статистичного аналізу, тому цей продукт має безліч вбудованих функцій для складних статистичних обчислень, побудови графіків та аналізу великих наборів даних. Його активно використовують у біостатистиці, екології тощо (Borcard et al., 2018). Python, навпаки, є універсальною мовою програмування з потужною системою бібліотек для машинного навчання, обробки тексту, веб-розробки та автоматизації (McKinney, 2018). Ми використовували у нашій методиці середовище розробки PyCharm, що є одним з найпопулярніших, проте не єдиним середовищем для використання мови Python (Komodo_IDE, VS Code, R, Spyder та інші) (McKinney, 2018). Вибір саме мови Python та середовища розробки PyCharm пов'язаний з нашим більшим досвідом роботи в них. Гіпотетично, аналогічні задачі можна виконувати у мові R, але функції та команди будуть відрізнятися через різний синтаксис мов. Альтернативою автоматизованому підходу є експорт Darwin Core Archive з Turboveg 2.0 (Hennekens, 2017), проте це також потребує додаткових дій з файлами.

В умовах України, де значна частина дослідницької інфраструктури постраждала через воєнні дії, автоматизація процесів узагальнення та обробки даних набуває виняткової актуальності. Водночас, у довгостроковій перспективі автоматизація сприятиме розвитку сучасної ботанічної інформатики в Україні, зростанню міжнародної співпраці та загалом посиленню наукової спроможності. Таким чином, метою роботи стало забезпечення перетворення геоботанічних описів у Long Data Format за допомогою мови Python відповідно до Darwin Core стандартів для наступної публікації Sampling-event набору даних у глобальній інформаційній мережі з біорізноманіття GBIF.

Матеріали і методи досліджень

Матеріалами слугували 1882 геоботанічних описів, виконаних на території Біосферного заповідника "Асканія-Нова" та у прилеглих цілинних подах Каховського і Генічеського р-нів Херсонської обл., Мелітопольського р-ну Запорізької обл. Часовий інтервал досліджень – 1967–2019 рр. Геоботанічні описи виконано за класичною методикою на постійних пробних площах екологічного ряду природного ядра Біосферного заповідника "Асканія-Нова", відновлених багаторічних перелогах, під час геоботанічних картувань та моніторингових обстежень за програмою "Літопис природи", у ході експедиційних досліджень регіону. Переважна більшість описів мали стандартну площу 100 м² (10×10 м), окремі описи обмежені площею від 9 до 16 м² (3×3 та 4×4 м відповідно). У вихідному наборі даних – електронній таблиці Excel – зведено інформацію про загальне та осібне проективне покриття видів рослин описової ділянки (різні шкали та бальні оцінки за потреби можуть бути уніфіковані до стандартного показника у %). Локалізація геоботанічних описів за період 2003–2019 рр. визначена за допомогою GPS-навігаторів Lowrance iFinder та Garmin eTrex 20x. Координати архівних описів наведені приблизно (з точністю ±500 м), за оригінальною характеристикою їх місцезнаходження у розрізі цілинних масивів та квартальної мережі природного ядра.

Власне, стаття зосереджена на роботі з розширенням Occurrence для Sampling-event набору даних та заповненні термінів до scientificName, eventID, occurrenceID, basisOfRecord, identifiedBy та recordedBy, які є обов'язковими для цього типу набору даних (Wieczorek et al., 2012). Основним методом для перетворення даних обрано мову Python, бібліотеку pandas (McKinney, 2018; Python Software Foundation, 2024). Коментування у скрипті виконано за допомогою ChatGPT 3.0. Розширення Relevé та Event-core здійснено за допомогою інструменту транспозиції у Excel-таблиці відповідно до термінів Darwin Core. Скрипт та вихідні дані опубліковано у відкритому доступі на GitHub (https://github.com/NadiiaSkobel/GBIF_LDT_usage_KSU), набір геоботанічних даних оприлюднено у мережі GBIF (Shapoval, Skobel, Vasyliuk, 2024).

Результати досліджень та їх обговорення

Трансформований набір даних по геоботанічних описах Біосферного заповідника "Асканія-Нова" та його регіону налічує 46839 реєстрацій 342 видів фітобіоти. Нижче описано апробований алгоритм перетворень.

Початковим етапом роботи є підготовка набору даних для імпорту у середовище розробки PycharmProjects (McKinney, 2018; Python Software Foundation, 2024). Щоб підготувати Excel-таблицю для роботи з Python, необхідно правильно структурувати дані (рис. 1). Заголовки таблиці мають бути на першому рядку, унікальними, без порожніх клітинок, із чіткими назвами, наприклад: Date, Quantity тощо. Важливо забезпечити однаковий тип даних у кожній колонці: дати мають бути у відповідному форматі дати, числа без текстових символів, а текст без зайвих пробілів. Таблиця не повинна містити об'єднаних клітинок, побічного форматування, коментарів чи графіків. Після цього файл потрібно зберегти у форматі .xlsx або .csv. Правильна підготовка таблиці спростить наступну роботу з даними та зменшить ймовірність помилок (McKinney, 2018).

Порядок дій для виконання скрипту:

1. Встановити PyCharm, середовище розробки для Python.

Завантажте інсталятор PyCharm. Перейдіть на офіційний сайт PyCharm (<https://www.jetbrains.com/pycharm/>) і завантажте версію для вашої операційної системи (Windows, macOS, Linux). Community Edition – рекомендована безкоштовна версія для базового використання; Professional Edition – платна версія з розширеними функціями для роботи з веброзробкою, наукою про дані та іншими технологіями.

2. Встановити на Windows.

Запустіть завантажений .exe файл. Виконайте інструкції майстра встановлення. Виберіть папку для встановлення. На етапі вибору опцій, відзначте: "Create Desktop Shortcut" (створення ярлика), "Add 'Open Folder as Project'" (додати контекстне меню в Провіднику Windows). Натисніть Install. Дочекайтеся завершення встановлення та натисніть Finish. Перезапустіть комп'ютер після інсталяції.

```

import pandas as pd

# Завантажте таблицю
file_path = 'askaniia.xlsx' # Вкажіть шлях до вашого Excel-файлу

# Завантаження даних, враховуючи, що заголовки починаються з 11-го рядка
df = pd.read_excel(file_path, header=10)

# Перетворення даних у формат long
long_df = (
df.melt(id_vars=df.columns[0], # Ідентифікаційні змінні - перша колонка
var_name='eventID',
value_name='organismQuantity')
.dropna(subset=['organismQuantity']) # Залишаємо лише непорожні комірки в колонці 'organismQuantity'
)

# Перетворення колонки 'eventID' у рядковий тип
long_df['eventID'] = long_df['eventID'].astype(str)

# Видаляємо суфікси типу .1, .2 тощо з колонок дат у 'eventID'
long_df['eventID'] = long_df['eventID'].str.replace(r'\.d+$', "", regex=True)

# Додаємо унікальні ідентифікатори 'occurrenceID' для кожного eventID
long_df['occurrenceID'] = long_df.groupby('eventID').cumcount() + 1
long_df['occurrenceID'] = long_df.apply(
lambda x: f"{x['eventID']}_{x['occurrenceID']:03}", axis=1
)

# Додаємо префікс до кожного значення в колонці 'eventID'
long_df['eventID'] = 'Askania_Nova_' + long_df['eventID'].astype(str)

# Додаємо префікс до кожного значення в колонці 'occurrenceID'
long_df['occurrenceID'] = 'Askania_Nova_' + long_df['occurrenceID'].astype(str)

# Додаємо нову колонку 'basisOfRecord' з постійним значенням 'HumanObservation' та ін.
long_df['basisOfRecord'] = 'HumanObservation'

# Зберігаємо у новий файл або виводимо результат
long_df.to_csv('long_data_format.csv', index=False)

```

Рисунок 1. Приклад коду у середовищі PyCharm.

Figure 1. Code example in the PyCharm environment.

3. Помістити робочий документ у робочу папку.

Наприклад C:\Users\skobe\PycharmProjects\pythonProject. Рекомендовано не використовувати кирилицю у шляху та назві файлу\скрипту для уникнення помилки при зчитуванні скрипту.

4. Встановити та імпортувати бібліотеку Python у терміналі pandas: для обробки даних, зокрема завантаження Excel-файлів, маніпулювання даними в таблицях, і збереження результату у CSV.

5. Скопіювати код вище або завантажити скрипт та вихідні дані з https://github.com/NadiiaSkobel/GBIF_LDT_usage_KSU, помістити в папку PycharmProjects та запустити скрипт.

6. Встановити необхідні бібліотеки.

Використайте такі команди для їх встановлення у терміналі: `pip install pandas`.

Після виконання указаних дій ви зможете працювати зі скриптом у середовищі розробки Pycharm та задавати свої задачі та функції. Натисніть правою кнопкою миші на файл і оберіть **Run 'script'** або натисніть **Shift + F10** для запуску. Повідомлення у терміналі **"Process finished with exit code 0"** свідчить про успішне виконання, а **"Process finished with exit code 1"** вказує на помилку. У середовищі PyCharm також подається інформація про помилку та зазначається в якій лінії коду вона наявна (рис. 2).

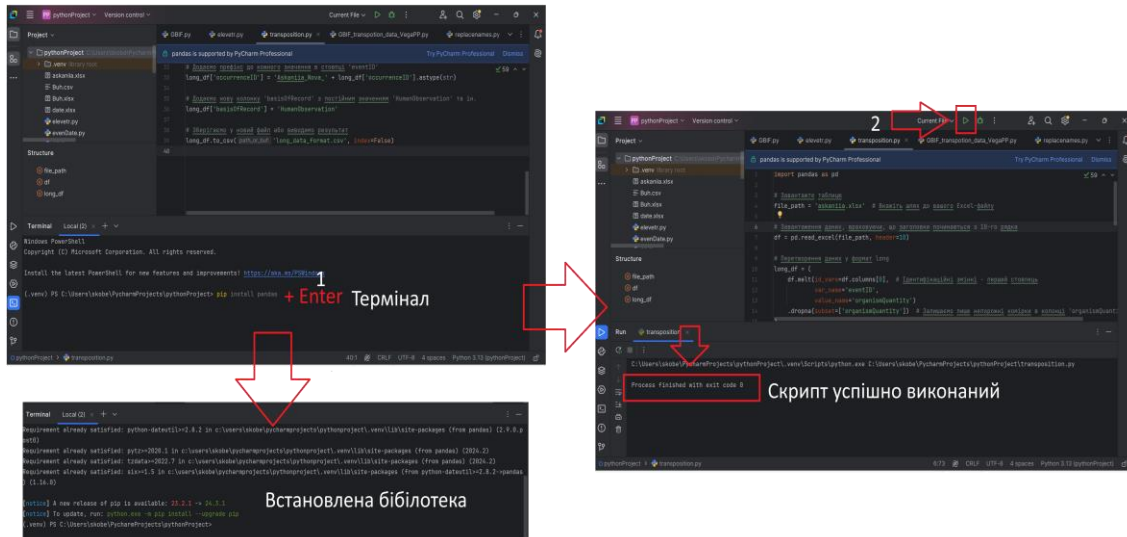


Рисунок 2. Узагальнена схема виконання дій у середовищі PyCharm: встановлення необхідної бібліотеки у Терміналі (1) та запуск скрипту (2).

Figure 2. A general workflow in the PyCharm environment: installing the necessary library through the Terminal (1) and running the script (2).

Опис функцій, які використовуються в коді, та аргументи, що вказують як завантажити, обробити та зберегти дані у відповідному форматі, залежно від структури початкового файлу та вимог до кінцевого результату, подано у таблиці 1. Приклад вихідних даних геоботанічного набору у Wide Data Format та результат його перетворення у Long Data Format представлено на рисунку 3.

Таблиця 1. Опис функцій та аргументів

Table 1: Description of functions and arguments

Функція	Призначення	Аргументи	Результат
1	2	3	4
pd.read_excel	завантажує дані з Excel-файлу у DataFrame	file_path: шлях до Excel-файлу; header=10: вказує, що заголовки колонок починаються з 11-го рядка; NOTE: Нумерація в мовах програмування починається з 0. Загальні дані представлені в хедерах в 11 рядках. Дані про номер опису розташовано нами в 11 рядку. Це індивідуально для кожного набору даних. Наш код пов'язаний з однією змінною – номером опису, тому ми ігнорували дані представлені вище. Для заповнення identifiedBy та recordedBy ми замінили дані в 11 рядку на Author code. Теоретично можна поєднати і кілька змінних	створює DataFrame із завантажених даних
melt	перетворює DataFrame із широкого формату в довгий формат	id_vars: колонки, які залишаються без змін (ідентифікаційні змінні, у вашому випадку перша колонка); var_name: назва нової колонки, яка міститиме назви попередніх колонок; value_name: назва нової колонки, яка міститиме значення з попередніх колонок	створює DataFrame у довгому форматі
dropna	видаляє рядки з порожніми значеннями у вказаних колонках	subset=['organismQuantity']: колонка, в якій перевіряються значення; NOTE: Важливо прибрати "." або "NaN" тощо з описів, як позначки пустих видів, якщо вони присутні. Альтернативою може бути команда на ігнорування подібних значень. Як результат, ці значення зчитуються як непорожні і будуть захищені у довгий формат	створює DataFrame без рядків із порожніми значеннями в колонці organismQuantity
astype	змінює тип даних у колонці	'str': перетворює значення у рядковий тип	створює оновлену колонку із вказаним типом даних

Кінець таблиці 1

1	2	3	4
str.replace	замінює частину тексту в значеннях колонки	r'\. \d+\$': регулярний вираз, який знаходить суфікси типу .1, .2 тощо; ": значення, на яке замінюється текст; regex=True: дозволяє використовувати регулярні вирази	створює колонку без небажаних суфіксів
groupby	групує дані за певною колонкою	'eventID': колонка, за якою здійснюється групування	створює об'єкт групування, що використовується для подальших обчислень
cumcount	нумерує кожен елемент у межах групи, від 0	-	створює послідовний індекс для кожної групи
apply	застосовує функцію до кожного рядка або значення в колонці	lambda x: f"{x['eventID']}_{x['occurrenceID']:03}": створює унікальний ідентифікатор для кожного рядка	створює нові значення у колонці
long_df	створює нову колонку і заповнює її однаковим значенням	-	створює нову колонку і заповнює її однаковими значеннями
to_csv	зберігає DataFrame у файл формату CSV	long_data_format.csv': назва файлу для збереження; index=False: виключає індекси DataFrame зі збереженого файлу	створює CSV-файл із обробленими даними

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Table number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	Country code	UA	UA	UA	UA	UA	UA	UA	UA	UA	UA	UA	UA	UA	UA	UA	UA
3	Author code	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38
4	Date (year/month/day)	20140922	20140922	20090605	20090605	20090605	20090605	20090606	20090606	20090606	20090606	20090606	20090606	20090606	20090606	20090606	20090606
5	RelevA area (m2)	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
6	Cover total (%)	85	90	70	60	50	80	60	40	65	70	70	70	85	80	85	80
10	Cover moss layer (%)																
11	Cover lichen layer (%)																
13	Cover litter layer (%)			5					15	10	10						10
16	Remarks	Pivdenna s Pivdenna s Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna : Pivnichna :															
17	latitude	46.457033	46.45705	46.487103	46.487191	46.487321	46.487731	46.482074	46.484467	46.484443	46.485044	46.488133	46.490267	46.4913	46.488517	46.48818	46.48800
18	longitude	34.102683	34.10225	33.964699	33.965718	33.966919	33.970301	34.035655	34.034750	34.03349	34.03280	34.02995	34.0258	34.024783	34.02395	34.02334	34.02209
19	kvartal	75	75	13	13	13	13	24	24	24	24	24	24	17	23	23	23
26	Achillea micranthoides	r	1
36	Allium pascoskianum	r
38	Allium regelianum	r	r
52	Androsace elongata	2	1	.
54	Anisantha tectorum	+
56	Arabidopsis thaliana	1	1	1	.	2	1
58	Arenaria uralensis	1	.	.	1	.
60	Artemisia austriaca	r	r	1	3	3	1	1	.	.	1	2	2	1	2	1	.
76	Bellevalia sarmatica	1
92	Carduus uncinatus	.	.	r	r	+	r	1	+	.	+	.	+	+	+	r	r

a

Table number	eventID	organismQuantity	occurrenceID	basisOfRecord
Achillea micranthoides	Askaniaia_Nova_1	r	Askaniaia_Nova_1_001	HumanObservation
Allium pascoskianum	Askaniaia_Nova_1	r	Askaniaia_Nova_1_002	HumanObservation
Artemisia austriaca	Askaniaia_Nova_1	r	Askaniaia_Nova_1_003	HumanObservation
Bromopsis inermis	Askaniaia_Nova_1	2	Askaniaia_Nova_1_004	HumanObservation
Buglossoides arvensis	Askaniaia_Nova_1	r	Askaniaia_Nova_1_005	HumanObservation
Carex melanostachya	Askaniaia_Nova_1	r	Askaniaia_Nova_1_006	HumanObservation
Carex praecox	Askaniaia_Nova_1	3	Askaniaia_Nova_1_007	HumanObservation
Centaurea diffusa	Askaniaia_Nova_1	r	Askaniaia_Nova_1_008	HumanObservation
Convulvulus arvensis	Askaniaia_Nova_1	r	Askaniaia_Nova_1_009	HumanObservation
Elytrigia repens	Askaniaia_Nova_1	r	Askaniaia_Nova_1_010	HumanObservation
Eryngium campestre	Askaniaia_Nova_1	r	Askaniaia_Nova_1_011	HumanObservation
Eryngium planum	Askaniaia_Nova_1	r	Askaniaia_Nova_1_012	HumanObservation
Euphorbia virgata	Askaniaia_Nova_1	r	Askaniaia_Nova_1_013	HumanObservation
Falcaria vulgaris	Askaniaia_Nova_1	r	Askaniaia_Nova_1_014	HumanObservation
Festuca valesiaca	Askaniaia_Nova_1	r	Askaniaia_Nova_1_015	HumanObservation
Galium ruthenicum	Askaniaia_Nova_1	r	Askaniaia_Nova_1_016	HumanObservation
Lepidium perfoliatum	Askaniaia_Nova_1	r	Askaniaia_Nova_1_017	HumanObservation
Limonium sareptanum	Askaniaia_Nova_1	r	Askaniaia_Nova_1_018	HumanObservation
Phlomis hybrida	Askaniaia_Nova_1	r	Askaniaia_Nova_1_019	HumanObservation
Phlomis scythica	Askaniaia_Nova_1	r	Askaniaia_Nova_1_020	HumanObservation
Poa angustifolia	Askaniaia_Nova_1	3	Askaniaia_Nova_1_021	HumanObservation
Potentilla argentea	Askaniaia_Nova_1	+	Askaniaia_Nova_1_022	HumanObservation

b

Рисунок 3. Вихідні дані геоботанічного набору у Wide Data Format (a) та результат їх перетворення у Long Data Format (b).

Figure 3. Source vegetation plots (relevés) dataset in Wide Data Format (a) and the result of its conversion to Long Data Format (b).

Висновки

Розробка та застосування скриптів Python у середовищі розробки PyCharm для перетворення геоботанічних описів у формат Long Data відповідно до стандартів Darwin Core є важливим кроком у інтеграції даних для публікації у глобальній інформаційній мережі з біорізноманіття GBIF. Використання мов програмування забезпечує автоматизацію процесу обробки великих обсягів геоботанічних даних, мінімізує похибки, немінучі при переведенні даних ручним способом, значно збільшує їх достовірність та доступність. Однак, як і в будь-якому методі, існують певні обмеження, зокрема пов'язані з необхідністю базових знань програмування, очищення даних (як необхідний підготовчий етап перед імпортом у середовище розробки), а також потенційними труднощами у налаштуванні скриптів для специфічних наборів даних. Водночас досягнення такого рівня автоматизації в обробці даних відкриває нові можливості для розвитку ботанічних інформаційних технологій в Україні, збереження біорізноманіття та даних про нього.

Подяки

Скобель Надія висловлює щирі подяки за фінансову підтримку досліджень Ukraine Future Leaders Program.

- Borcard, D., Gillet, F., & Legendre, P. (2018). *Numerical Ecology with R* (2nd ed.). Springer. <https://doi.org/10.1007/978-3-319-71404-2>
- Hennekens, S.M. (2017). *Turboveg for Windows* (version 1998–2017). 100 pp.
- McKinney, W. (2018). *Python for Data Analysis* (3rd ed.). *O'Reilly Media*. 536 pp.
- Python Software Foundation. (2024). *The Python Programming Language*. Python.org. <https://www.python.org/>
- R Core Team (2024) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>
- Shapoval V, Skobel N, Vasyliuk O (2024). Records of vascular plants, bryophytes and lichens in the core of the Askania-Nova Biosphere Reserve F.E. Falz-Fein and depressions lands based on relevés in 1967–2019. Version 1.1. Ukrainian Nature Conservation Group (NGO). Sampling event dataset <https://doi.org/10.15468/m4u6mx>
- Wieczorek J, Bloom D, Guralnick R, Blum S, Döring M, et al. (2012) Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. *PLoS ONE* 7(1): e29715. <https://doi.org/10.1371/journal.pone.0029715>

Received: 9 October 2024 / Revised: 17 November 2024 / Accepted: 30 December 2024