Supplementary Information—
Innovation in electric secondary batteries: Patenting trends in technologies and geographic origins

Philipp Metzger[1,*], **Bruno Damásio**[1], **José A. Silva**[2], **and Sandro Mendonça**[3]

[1]NOVA IMS Information Management School, Lisbon, Portugal
[2]Instituto Dom Luiz, Faculdade de Ciências Universidade de Lisboa, Lisbon, Portugal
[3]ISCTE Business School, Business Research Unit (BRU-IUL), Lisbon, Portugal
[*]philipp.metzger.bat.pat@gmail.com

Battery IPF intensities for each continent

Figure S1 presents the development of the number of battery IPFs per 1M workers (battery IPF intensities) for each continent. In terms of battery IPF intensities, Europe and North America outperform Asia. Asia contributed approximately 60% to the global labor force in the timeframe of 2000-2019 (Europe and North America contributed approximately 9% and 8%, respectively). This imbalance explains why Asia's battery IPF intensities are lower in the perspective of this representation.

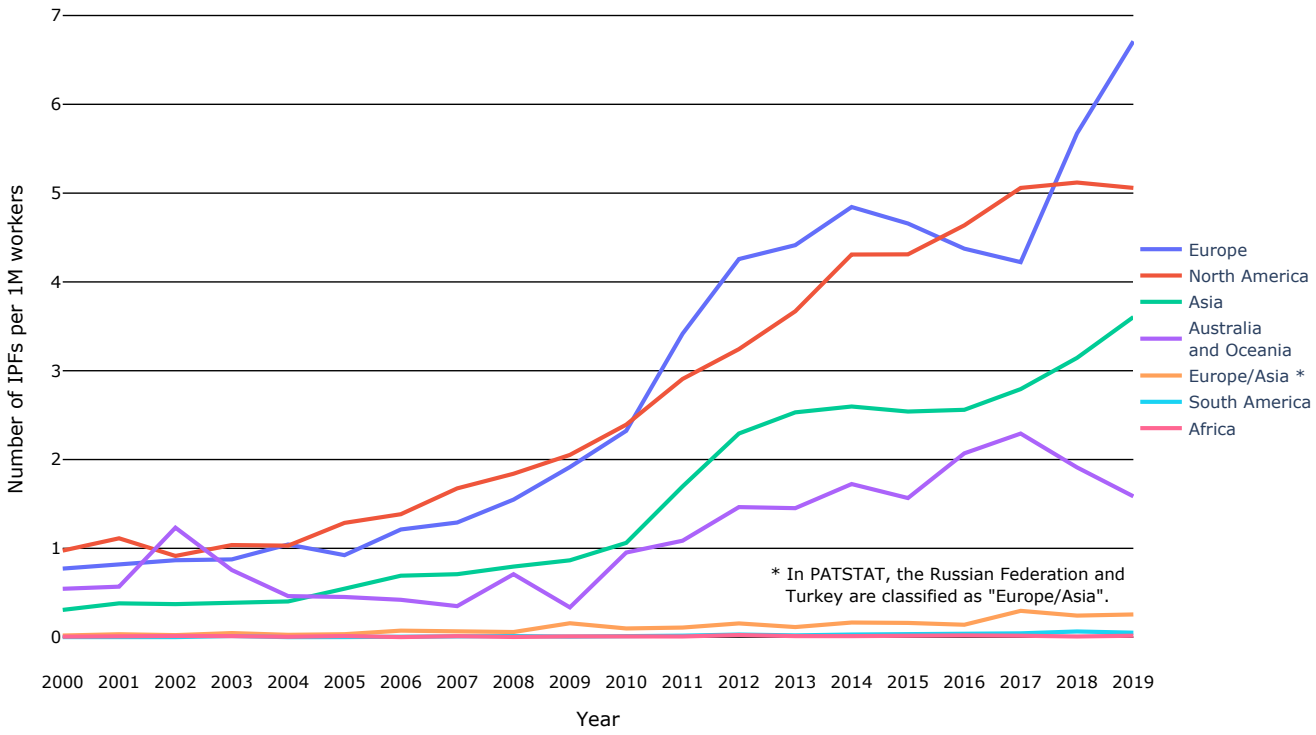

Figure S1. Development of the number of battery IPFs per 1M workers (intensities): Counted by inventors' continents of origin, 2000-2019. In terms of battery IPF intensities, Europe and North America outperform Asia.

Python packages

The programming language used for all steps after querying the PATSTAT database and downloading the data was Python[1] (Version 3.9.7). The packages used for this study are the web application Jupyter Notebook[2] (Version 6.4.3), the data processing libraries pandas[3] (Version 1.3.3) and Numpy[4] (Version 1.20.3), the visualisation tools Plotly[5] (Version 5.1.0) and Seaborn[6] (Version 0.11.2), the text mining suite Natural Language Toolkit (NLTK)[7] (Version 3.6.5) and the analytics toolboxes

Scikit-learn[8] (Version 0.24.2) and SciPy[9] (Version 1.7.1).

Detailed description of preprocessing and data reduction steps

First, the raw data downloaded from PATSTAT Online was loaded and checked for its integrity. Then each patent family's earliest intra-family values for the features "earliest publication date" and "earliest publication year" were determined and added as a new columns to every row of the dataset (i.e. they were harmonized on patent family level). Like this, patent families can easily be assigned to their respective year later during the analyses. Next, all patent families were classified and tagged as either "IPF", "singleton", or "neither". The resulting tags are stored in the newly created column "tag". Next, more tags for further data selection were created. This process took place in five steps, which are described in the following:

- First, every patent family was scanned for the IPC codes related to non-active battery parts, electrodes, or secondary cells (IPC codes H01M 2..., H01M 50..., H01M 4..., and H01M 10...). Patent families that contained any of these code were added in their entirety, except if they contained any of the IPC codes H01M 6..., H01M 8..., H01M 12..., H01M 14..., or H01M 16..., which are related to primary cells, fuel cells, hybrid cells, electrochemical current or voltage generators not provided for in groups H01M 6/00-H01M 12/00, and structural combinations of different types of electrochemical generators, which were hereby explicitly excluded from the analysis. The patent families passing this stage were tagged as "non active parts, electrodes, secondary cells".

- In a second step, every patent family was scanned for the IPC codes related to "circuit arrangements for ac mains or ac distribution networks using batteries with converting means" (H02J 3/32), "circuit arrangements for charging or depolarising batteries or for supplying loads from batteries" (H02J 7...), "methods of charging batteries, specially adapted for electric vehicles" (B60L 53...), or "secondary cells; methods for charging or discharging" (H01M 10/44). Patent families that contained any of these codes were added in their entirety, except if they contained any of the IPC codes listed for exception in the above step or any of the codes B60L 53/54, B60L 53/55, or B60L 53/56 that refer to charging stations using fuel cells, capacitors, or mechanical storage means, respectively. Patent families that passed this stage were tagged as "charging".

- As a third step, in order to identify affiliations of the resulting patent families to a set of technological categories, each patent family's titles and abstracts were scanned using individual sets of regular expressions for each technology. These regular expressions are defined in the Jupyter notebook "01_create_dataset.ipynb". Titles and abstracts of all languages were considered and a patent family was selected in its entirety if any substring of its titles or abstracts matched any of the respective regular expressions. Please note that—in order to decrease the risk of false positives—before scanning abstracts for these regular expressions, they were cut off at the beginning of any appearance of the string "independent claims are also included for". The selected patent families were one-hot tagged in the newly created columns with the column name "is x", with $x \in$ {Lead-acid, Lithium-air, Lithium-ion, Lithium-sulfur, Other Lithium, Magnesium-ion, Nickel-cadmium, Nickel-iron, Nickel-zinc, Nickel-metal hydride, Rechargeable alkaline, Sodium-sulfur, Sodium-ion, Solid-state, Aluminium-ion, Calcium(-ion), Organic radical} being the name of the respective technology. Please note that due to the considerable overlap of the concept of solid-state batteries with other technologies, especially lithium-ion batteries, all patent families that were classified as patents related to solid-state batteries were untagged in any other category in which they acquired tags through the process described here. To be very clear: This especially means that the lithium-ion battery category does not contain any patent families that are tagged as solid-state battery inventions.

- The fourth step's purpose was to add patent data related to redox flow and nickel–hydrogen batteries to the dataset. For this purpose, a combination of IPC classes queries and text queries was deployed. The reason for this separate step is that redox flow and nickel–hydrogen batteries are closely related to fuel cells and, consequently, patents associated with them are often included in IPC classes that were excluded by the above steps. Analogous to the above steps, the IPC classes qualifying for potential inclusion were H01M 2..., H01M 50..., H01M 4..., H01M 8..., and H01M 10... and the IPC classes demanding exclusion were H01M 6..., H01M 12..., H01M 14..., and H01M 16.... Analogous to the above step, these patent families' titles and abstract were then scanned using one set of regular expressions for redox flow and another for nickel–hydrogen batteries. These regular expressions can be reviewed in the Jupyter notebook "01_create_dataset.ipynb". All patent families that passed this stage were one-hot tagged in the newly created columns with the names "is Redox flow" or "is Nickel–hydrogen", respectively.

- As a last step, another additional column was computed: The dataset column "technologies one hot sum" contains the sum across each row's "is <technology name>" values. This sum is needed in the rare cases where technology classifications overlap. The share of patent families that had more than one technology associated to them was 0.61% in the final dataset.

The counts resulting from these overlapping technologies were not counted multiple times but, using the respective "technologies one hot sum" value, distributed as equal fractions across the overlapping classes.

The tags created in the above steps were used for selecting the appropriate data for each analysis. All patent families not having the "IPF" tag were filtered out before all analyses. That the rest was kept in the unfiltered dataset was only for completeness, having potential future analyses with a broader scope in mind. The data selection method that was applied before each analysis that is based on the labels whose creation was described above is presented in Fig. S2.

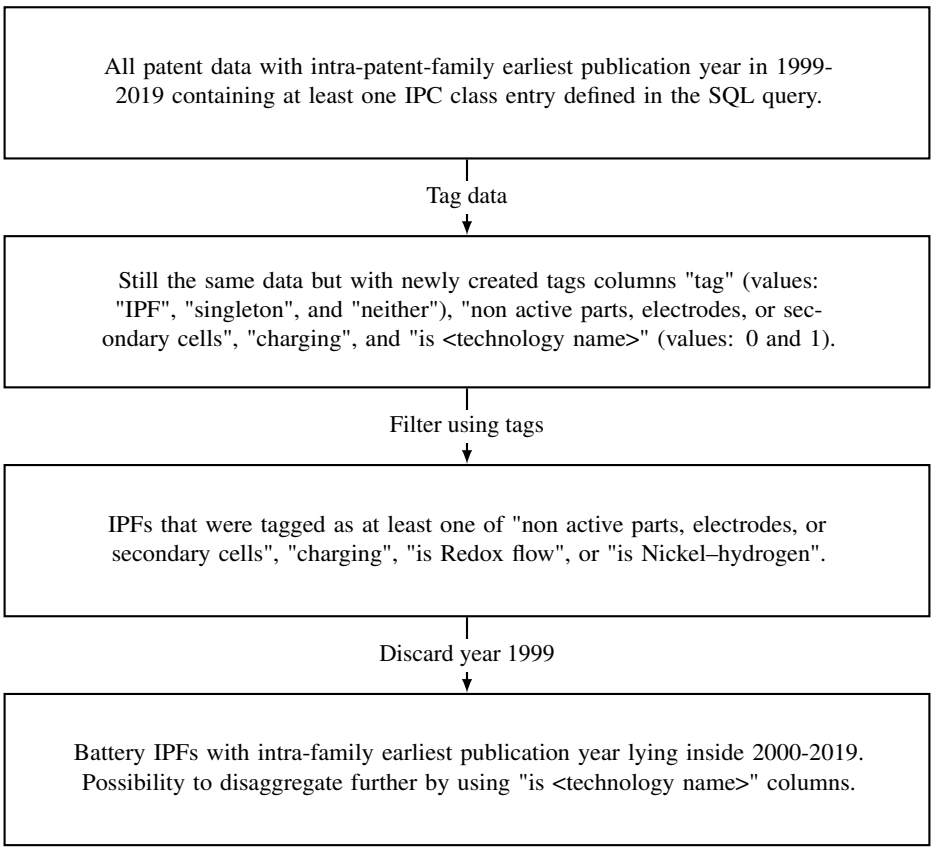

Figure S2. Flow chart depicting the data selection process for this study. The entire dataset raw dataset was labeled using newly created columns. Before each analysis, the final dataset was acquired by filtering using labels and timestamp columns.

Characterization of R^2

The measure R^2 used for comparing the performance of several clustering algorithms using varying numbers of clusters can be characterized as follows—please note that said comparison was conducted on a non-varying dataset:

$$R^2 = \frac{SSB}{SST} = \frac{SST - SSW}{SST} = 1 - \frac{SSW}{SST} \in [0,1] \tag{1}$$

where

$$SSB = \sum_{i=1}^{p} n_i (\overline{X}_i - \overline{X})^2 = sum\ of\ squared\ differences\ between\ groups \tag{2}$$

and

$$SSW = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})^2 = sum\ of\ squared\ differences\ within\ groups \tag{3}$$

and

$$SST = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X})^2 = total\ sum\ of\ squared\ differences \tag{4}$$

with

$p = number\ of\ clusters,$

$n_i = number\ of\ elements\ in\ cluster\ i,$

$\overline{X_i} = centroid\ of\ cluster\ i,$

$\overline{X} = center\ of\ whole\ dataset,\ and$

$X_{ij} = jth\ element\ of\ cluster\ i.$

Relations 1 are true if and only if $SST = SSW + SSB$, which is the case because:

$$SST = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X})^2 = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} \underbrace{-\overline{X_i} + \overline{X_i}}_{=0} - \overline{X})^2 = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})^2 + \sum_{i=1}^{p} \sum_{j=1}^{n_i} (\overline{X_i} - \overline{X})^2 + 2 \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})(\overline{X_i} - \overline{X})$$

$$= \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})^2 + \sum_{i=1}^{p} \sum_{j=1}^{n_i} (\overline{X_i} - \overline{X})^2 + 2 \sum_{i=1}^{p} (\overline{X_i} - \overline{X}) \underbrace{\overbrace{\sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})}^{(5)}}_{=0} = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})^2 + \sum_{i=1}^{p} \sum_{j=1}^{n_i} (\overline{X_i} - \overline{X})^2$$

$$= \overbrace{\sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})^2}^{(3)} + \overbrace{\sum_{i=1}^{p} n_i (\overline{X_i} - \overline{X})^2}^{(2)} = SSW + SSB$$

with

$$\sum_{j=1}^{n_i} (X_{ij} - \overline{X_i}) = \sum_{j=1}^{n_i} X_{ij} - \sum_{j=1}^{n_i} \overline{X_i} = \frac{n_i}{n_i} \sum_{j=1}^{n_i} X_{ij} - n_i \overline{X_i} = n_i \overline{X_i} - n_i \overline{X_i} = 0 \tag{5}$$

References

1. Van Rossum, G. & Drake, F. L. *Python 3 Reference Manual* (CreateSpace, Scotts Valley, CA, 2009).
2. Kluyver, T. *et al.* Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. & Schmidt, B. (eds.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87 – 90 (IOS Press, 2016).
3. Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (eds.) *Proceedings of the 9th Python in Science Conference*, 56 – 61, DOI: 10.25080/Majora-92bf1922-00a (2010).
4. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362, DOI: 10.1038/s41586-020-2649-2 (2020).
5. Inc., P. T. Collaborative data science (2015).
6. Waskom, M. *et al.* mwaskom/seaborn: v0.11.2 (august 2021), DOI: 10.5281/zenodo.5205191 (2021).
7. Bird, S., Klein, E. & Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit* (" O'Reilly Media, Inc.", 2009).
8. Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
9. Virtanen, P. *et al.* Scipy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods* **17**, 261–272, DOI: 10.1038/s41592-019-0686-2 (2020).