MakeItSample: a Python Library for Generating Typological Language Samples Based on the Diversity Value Metric

Luca Brigada Villa

Dipartimento di Studi Umanistici, Università di Pavia, Piazza del Lino, 2 - 27100 - Pavia, Italy

Abstract

This paper presents `makeitsample`, a Python library for generating typological language samples based on the diversity value (DV) metric. The library handles the construction of hierarchical language family trees from a list of CSV, the calculation of diversity values for each node in the trees, and the selection of languages based on their weight within the tree. The library aims to ease the process of creating typological language samples by providing an automated, scalable, and reproducible solution.

Keywords

typology, sampling, diversity value, language family tree, typological databases

1. Introduction

Linguistic typology is the study of structural patterns and variation across the world's languages [1, 2]. Since there are over 7,000 known languages [3], full coverage of linguistic diversity in typological studies is unfeasible. Instead, researchers rely on language samples — subsets of languages selected to represent the world's linguistic diversity as accurately as possible [4, 5]. However, the way these samples are constructed greatly impacts the validity of typological generalizations, as biased sampling can distort conclusions about universal tendencies and linguistic variation [6].

Several sampling strategies have been developed to improve representativeness in typological studies. Random sampling is a straightforward method, but it risks including many closely related languages, reducing genealogical and areal diversity [5, 7]. Stratified sampling mitigates this issue by ensuring balanced representation across language families and geographic regions [8], yet defining appropriate strata remains a challenge. For instance, genealogical classification varies between databases such as Glottolog [3] and Ethnologue [9], leading to inconsistencies in sampling.

Another approach is diversity-based sampling, which prioritizes structurally diverse languages rather than simply ensuring equal representation across language families or regions [6]. This method focuses on maximizing linguistic variation within a sample, making it particularly useful for detecting cross-linguistic patterns [10]. While promising, current implementations of diversity-based sampling often lack computational automation and clear reproducibility, limiting their practical application.

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy

luca.brigadavilla@unipv.it (L. Brigada Villa)

0009-0003-3523-7622 (L. Brigada Villa)

Despite efforts to refine sampling methods, typological research remains susceptible to several biases [11]:

- Bibliographic bias: since typological studies rely on existing descriptions, well-documented languages are favored over lesser-described or endangered languages [12]. In addition to this, the quality of the descriptions may affect the results of the typological analysis, as some grammars may have been written with a specific theoretical framework in mind, or been written in the past and not updated to reflect current linguistic theories.
- Genetic bias: samples may be unbalanced due to the overrepresentation of some language families, leading to an underestimation of linguistic diversity [4, 7].
- Areal bias: some geographic regions (e.g., Europe) are disproportionately represented in typological databases compared to highly diverse but underdocumented areas such as New Guinea and the Amazon [13, 14].
- Typological bias: this bias occurs when a sample contains a disproportionate number of languages with similar typological features, leading to overgeneralizations about linguistic universals [6]. For example, if a sample contains a large number of SVO languages, it may lead researchers to conclude that SVO is the most common word order across languages or that a feature associated with this order (e.g. adjective-noun order) is the most common across languages, even if this is not the case. This bias can also occur when researchers focus on a specific typological feature (e.g., case marking) and select languages that exhibit that feature.
- Cultural bias: this bias occurs when language samples underrepresent the world's cultural and

linguistic diversity. It relates to the idea of linguistic relativity—the notion that language can influence how people think and perceive the world [15, 16]. While early theories assumed a strong, deterministic link, more recent research treats the connection between language and thought as testable. For instance, Lucy [17] showed that speakers of languages with obligatory number marking perceive and categorize objects differently than speakers of classifier languages, illustrating how grammatical structures can reflect cultural patterns.

These biases can skew typological conclusions, reinforcing the need for an automated sampling pipeline that accounts for linguistic diversity in a principled manner.

To address one of these biases, this paper presents a Python library to ease the process of generating typological language samples. The library, called `makeitsample`[1], is designed to automate the sampling process and provide a principled and scalable solution to generating language samples for typological studies. The library implements a sampling method based on the diversity value (DV) metric [18, 19, 11] and comes with a command-line interface. The library is designed to:

- Construct a set of hierarchical language family trees from a set of CSV files.
- Compute diversity values (DVs) for each language family and subgroup, ensuring that more structurally diverse families contribute proportionally to the final sample.
- Select languages based on the weights of the groups and families they belong, propagating the selection algorithm from higher-level families down to subgroups, ensuring a genealogically and typologically balanced sample.

By integrating computational methods with linguistic typology, this library provides an automated, scalable, and genealogical bias-aware solution to sampling. The paper is structured as follows: Section 2 describes the methodology behind the DV metric and the sampling algorithm. Section 3 details the implementation of the package, describing the libraries it relies on and the modules of the library. Finally, Section 4 discusses the potential applications of `makeitsample` and concludes the paper.

[1]Available at https://pypi.org/project/makeitsample/. `makeitsample` is open source and licensed under the MIT license. The source code is available at https://github.com/unipv-larl/makeitsample.

2. Methodology

In this section, I describe the methodology behind the diversity value (DV) metric and the sampling algorithm. I first introduce the family tree representation used to model genetic relationships between languages (Section 2.1). Then, I explain how DVs are calculated for each language family and subgroup (Section 2.2). Finally, I detail the sampling algorithm that selects languages based on their weight within the tree (Section 2.3).

2.1. The Family Tree Representation

A family tree is a hierarchical structure that represents the genetic relationships between languages. Each node in the tree corresponds to a language family or subgroup, while edges indicate parent-child relationships. The hierarchical structure allows us to visualize the genealogical relationships between languages, with higher-level nodes representing broader families and lower-level nodes representing more specific subgroups or individual languages. This way of representing language families traces back to Schleicher's works [20, 21], where he proposed a tree-like structure to illustrate the relationships between languages. This representation has been widely adopted in historical linguistics and typology, as it provides a clear and intuitive way to visualize the genetic relationships between languages. The idea behind the family tree is to represent the evolution of languages over time, with branches representing the divergence of languages from their common ancestors. Each language family can be thought of as a trunk, with subgroups and individual languages branching out from it. The length of the branches can be interpreted as a measure of the time since the languages diverged from their common ancestor, with longer branches indicating greater divergence. The tree is rooted at the top-level family, with subgroups branching out from their respective parent nodes. This representation allows us to model the genealogical relationships between languages and determine their relative weights within the tree.

As an example, consider the Indo-European language family, which, according to Ethnologue [9], is divided into eight subgroups: Albanian, Armenian, Baltic, Celtic, Germanic, Greek, Indo-Iranian, and Italic. These subgroups are further divided into smaller subgroups and individual languages, forming a hierarchical structure that captures the genetic relationships between Indo-European languages as in Figure 1.

The family tree representation allows us to model the genetic relationships between languages and see which families and groups are more structurally diverse. This information is crucial for calculating diversity values and selecting languages for the final sample.

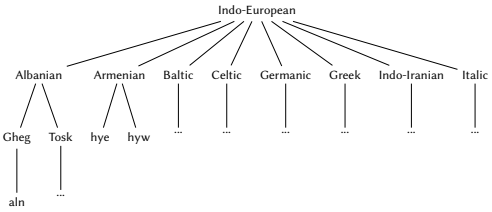

Figure 1: Sample of the tree of the Indo-European family. This representation does not take into account the temporal aspect of the tree, i.e. the length of the branches is not proportional to the time since the languages diverged from their common ancestor.

2.2. Calculating the Diversity Value (DV)

The diversity value (DV) metric quantifies the structural diversity of a language family or subgroup based on the topological properties of its family tree. This metric was first introduced by Rijkhoff and Bakker [18] and later refined by Bakker [11] as a way to maximize the typological diversity of languages in a sample. The calculation involves the following steps:

1. Breadth-First Search (BFS): starting from a given node for which we want to calculate the DV (henceforth "root"), perform a BFS to determine the level of each node in the tree. The level of a node is the number of edges from the root to that node.
2. Level Counts: calculate the number of nodes at each level. This helps in understanding the distribution of nodes across different levels of the tree.
3. Contributions Calculation: for each level, calculate the contributions to the DV. The contribution of a level is determined by the number of nodes at that level and their distance from the starting node. The contributions are accumulated as we move from the root to the leaves of the tree. The contribution C_i of level i can be calculated as:

$$C_i = C_{i-1} + (N_i - N_{i-1}) \times \frac{L - (i-1)}{L}$$

where C_{i-1} is the contribution of the level upwards (setting to 0 the contribution of the root level) N_i is the number of nodes at level i, N_{i-1} is the number of nodes at the level above, and L is the maximum number of levels in the forest. If we are calculating the DV for the root of the family tree, then L is the maximum number of levels in any tree in the forest. If we are calculating the DV for a subgroup, then L is the maximum number of levels in the sibling trees of the tree rooted at the subgroup (including the subgroup tree).

Sometimes, family trees are shaped like the left side tree in Figure 2 in which a branch of the tree stops at a certain level without reaching the bottom of the tree (see the group 1 branch in Figure 2). If we apply the previous formula, we would get a negative factor while calculating the contribution of the bottom level, since N_i would be lower than N_{i-1}. To avoid this, we add a number of pseudo-nodes to the tree (x nodes in Figure 2), so that the number of nodes at each level is always greater than or equal to the number of nodes at the level above. This is done by adding a number of pseudo-nodes equal to the difference between the number of nodes at the level above and the number of nodes at the current level. The pseudo-nodes are not included in the final sample, but they are necessary to ensure that the contributions are calculated correctly. The pseudo-nodes are added only to the levels that are not the last level of the tree. This way, we can ensure that the contributions are always positive and that the DV is calculated correctly.

4. Mean of Contributions: the DV is the mean of the contributions calculated in the previous step. This average value represents the structural diversity of the language family or subgroup. The DV can be expressed as:

$$DV = \frac{1}{D} \sum_{i=1}^{D} C_i$$

where D is the depth of the tree rooted at the node for which we are calculating the DV, and C_i is the contribution of level i.
For language isolates, the DV is set arbitrarily to 1 (as suggested by Rijkhoff and Bakker [19]), in order to avoid assigning a value of 0 to these languages and to ensure that they get the chance to be selected in the sampling algorithm.

By following these steps, we can compute the DV for any node in the family tree (except for nodes representing languages which are not structurally diverse in the tree). The DV metric provides a principled way to quantify the typological diversity of languages and guide the selection process in the sampling algorithm.

As a matter of example, let us consider the example forest in Figure 2 and let us suppose that we want to calculate the DV of the family 1. The first step is to define L, i.e. the maximum number of levels under the root node in the forest. In this case, $L = 3$. Then, we proceed to calculate the contributions of each level. For the first level, i.e. the one including group 1 and group 2, we have $N_1 = 2$ and $N_0 = 1$. C_0 is set to 0, so we have:

$$C_1 = 0 + (2-1) \times \frac{3-(1-1)}{3} = 0 + (1 \times 1) = 1.$$

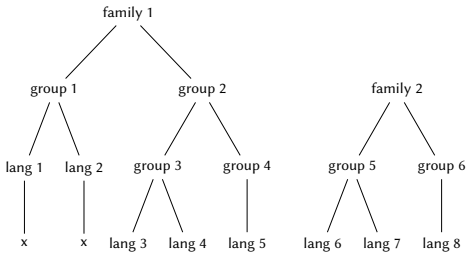

Figure 2: An example forest of language families.

For the second level, i.e. the one including lang 1, lang 2, group 3 and group 4, we have $N_2 = 4$ and $N_1 = 2$, so:

$$C_2 = 1 + (4-2) \times \frac{3-(2-1)}{3} = 1 + (2 \times \frac{2}{3}) = \frac{7}{3}.$$

For the third level, i.e. the one including the two pseudo-nodes, lang 3, lang 4 and lang 5, we have $N_3 = 5$ and $N_2 = 4$, so:

$$C_3 = \frac{7}{3} + (5-4) \times \frac{3-(3-1)}{3} = \frac{7}{3} + (1 \times \frac{1}{3}) = \frac{8}{3}.$$

Finally, we can calculate the DV as:

$$DV = \frac{1}{3}\left(1 + \frac{7}{3} + \frac{8}{3}\right) = \frac{1}{3} \times 6 = 2.$$

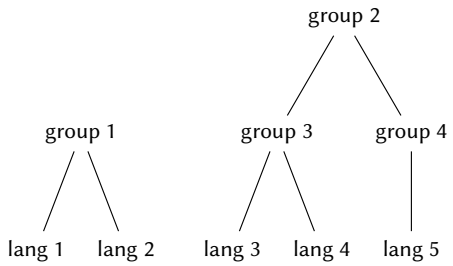

Figure 3: The forest obtained considering the sibling trees of group 1. The pseudo-nodes are not needed here since all the leaves are at the same level.

This algorithm can be applied to any node in the family tree. If we want to calculate the DV of a subgroup, we can simply set L to the maximum number of levels in the sibling trees of the tree rooted at the subgroup (including the subgroup tree). For example, if we want to calculate the DV of group 1, we can set $L = 2$ (since the maximum number of levels in the sibling trees is 2). Then, we can calculate the contributions as before, without considering the pseudo-nodes. The full calculation of the DV of this node and all the other nodes in the forest is not shown here for the sake of brevity, but it can be found in Appendix A.

2.3. The Sampling Algorithm

The sampling algorithm aims to select the most diverse set of languages from the family trees, ensuring that the final sample is representative of the world's linguistic diversity. Let us suppose that we need a sample of size N. If N is higher than the total number of languages in the family tree, we start by selecting at least a language from each family. If there is still a number of languages to be selected, we distribute this number among the families according to their DVs. The distribution is randomic but weighted by the DVs of the families. This ensures that more structurally diverse families contribute proportionally more to the final sample. If the sample size N is smaller than the total number of families, we select the families randomly, but weighted by their DVs and select a language from each selected family.

If the sample is not complete, we proceed selecting other languages. At this stage, each selected family has at least one language included in the sample. The remaining languages are then allocated to the subgroups of each family, continuing down to the individual language level. This allocation is done randomly but weighted by the diversity values of the nodes, as shown in Figure 4.

When each subgroup has been assigned a number of languages, we select the languages randomly from the subgroups.

3. Implementation

In this section, I describe the implementation of `makeitsample`, outlining the dependencies it utilizes (section 3.1), and the two modules of the library: `language_family_tree` (Section 3.2) and `forest` (Section 3.3). I also provide an overview of the command-line interface (Section 3.4) and the structure of the input data (Section 3.4.1).

3.1. Libraries

The modules rely mainly on two libraries: `pandas` [22, 23] for data manipulation and `networkx` [24] for graph representation and algorithms. The `pandas` library is used to read the input data and construct the family tree. The `networkx` library is used to represent the family tree as a graph and perform graph-based operations such as BFS traversal and DV calculation.

3.2. The `language_family_tree` Module

The `language_family_tree` module is responsible for constructing the family trees from the input data. It reads the CSV files and creates a hierarchical structure representing the genetic relationships between languages. It

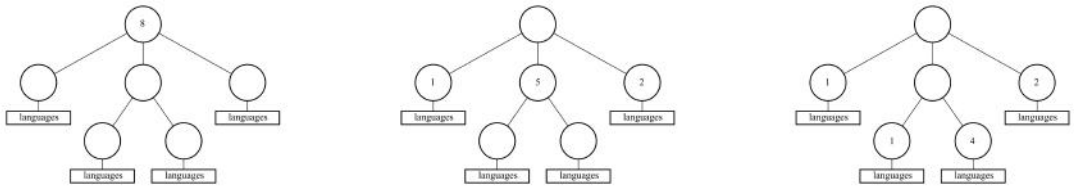

Figure 4: Illustration of the allocation to the subgroups. If we have to select 8 languages from this family tree (step 1), we start by selecting 1 language from each branch and distribute the remaining 5 languages among the branches (step 2). If we reach the bottom of the tree, we select the languages from the branch, otherwise we repeat the process (step 3).

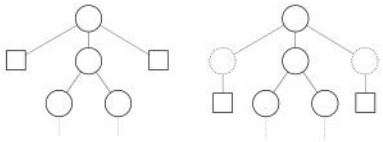

Figure 5: Example of the tree structure before and after adding a node. Circles represent subgroups and families, while squares represent languages.

consists of a class called `LanguageFamilyTree` inherited from the `networkx.DiGraph` class. This class represents the family tree as a directed graph, where each node corresponds to a language family, subgroup or language, and edges represent parent-child relationships. The class provides methods for building the tree from a CSV input (formatted as described in Section 3.4.1), for exporting the tree to a JSON or CSV file, for converting it to a dictionary, for calculating the diversity values of the nodes and for selecting a certain number of languages from the tree according to the sampling algorithm described in Section 2.3.

When importing the data, a function of `LanguageFamilyTree` refines the structure of the tree in order to avoid structures that would make impossible to be processed by the sampling algorithm. This occurs when a subgroup contains both languages and other subgroups as children. To address this, an additional level is introduced in the tree to separate the languages from the subgroups. This is achieved by creating new nodes that become parents to each language and children to the node that was previously their parent, as shown in Figure 5. This ensures the structure remains a tree, allowing the sampling algorithm to function correctly.

3.3. The `forest` Module

The `forest` module is responsible for managing multiple family trees and performing operations on them. It consists of a class called `Forest` that inherits from the `list` class. This class represents a collection of family trees and provides methods for reading a set of CSV files representing family trees from a directory, adding new `LanguageFamilyTree` objects to the forest, exporting the forest to a set of JSON or CSV files, calculating the diversity values of the trees in the forest, and selecting languages from the forest according to the sampling algorithm.

3.4. Command-Line Interface

The command-line interface (CLI) of `makeitsample` is designed to be user-friendly and allows users to easily run the sampling pipeline from the command line. To run the pipeline, users can use the following command:
`makeitsample [-h] [-n N] [-i INPUT] [-o OUTPUT] [-f {csv,json}] [-s SAMPLENAME] [-r RANDOM_SEED]`
where `N` is the sample size, `INPUT` is the input directory containing the CSV files, `OUTPUT` is the output directory where the sample will be saved, `f` is the output format (`csv` or `json`), `SAMPLENAME` is the name of the sample file, and `RANDOM_SEED` is the random seed for reproducibility.

3.4.1. Structure of the Input Data

In order to run `makeitsample`, the input data must be in a CSV format (as in the example in Table 1 in Appendix B). The CSV files (one for each language family) should contain:

- `id`: a column for the unique identifier of the language (e.g., ISO code), of the family or the group;
- `name`: a column storing the name of the language, of the family or the group;

- `parent_id`: a column storing the `id` of the parent node in the family;
- `type`: a column storing the type of the node (the only allowed values for this column are `family`, `group` or `language`).

The user can also add other columns with additional information about the languages, families or groups. `makeitsample` will ignore these columns when constructing the family tree, but they will be included in the output file.

4. Conclusions

In this paper, I presented `makeitsample`, a Python package that aims to ease the generation of typological language samples based on the diversity value (DV) metric. I presented the modules of the library and the command-line interface, which allow to construct a set of hierarchical language family trees, to calculate diversity values for each node, and to select languages based on their weight within the tree. By automating the sampling process and accounting for linguistic diversity, the library and the command-line interface provide a principled and scalable solution to generating language samples for typological studies helping researchers create more representative samples and reduce genealogical biases in their analyses.

The library is designed to be flexible and extensible, allowing researchers to adapt it to their specific needs and incorporate additional sampling strategies or metrics. Although user-friendly, the library is still in its early stages and requires some knowledge of Python to be used effectively or at least some familiarity with the command line. This might be a limitation for some users, and the plan is to create a web interface to make it more accessible to a wider audience.

References

[1] B. Comrie, Language Universals and Linguistic Typology: Syntax and Morphology, University of Chicago Press, Chicago, 1989.

[2] W. Croft, Typology and Universals, Cambridge Textbooks in Linguistics, 2 ed., Cambridge University Press, Cambridge, 2002.

[3] H. Hammarström, R. Forkel, M. Haspelmath, S. Bank, Glottolog 5.1, 2024. URL: http://glottolog.org. doi:10.5281/zenodo.14006617.

[4] M. S. Dryer, Large linguistic areas and language sampling, Studies in Language. International Journal sponsored by the Foundation "Foundations of Language" 13 (1989) 257–292. URL: https://www.jbe-platform.com/content/journals/10.1075/sl.13.2.03dry. doi:https://doi.org/10.1075/sl.13.2.03dry.

[5] R. D. Perkins, Statistical techniques for determining language sample size, Studies in Language. International Journal sponsored by the Foundation "Foundations of Language" 13 (1989) 293–315. URL: https://www.jbe-platform.com/content/journals/10.1075/sl.13.2.04per. doi:https://doi.org/10.1075/sl.13.2.04per.

[6] B. Bickel, Distributional typology: Statistical inquiries into the dynamics of linguistic diversity, in: B. Heine, H. Narrog (Eds.), The Oxford Handbook of Linguistic Analysis, 2nd ed., Oxford University Press, Oxford, 2015, pp. 901–923.

[7] J. Nichols, Linguistic Diversity in Space and Time, University of Chicago Press, Chicago, 1999.

[8] M. S. Dryer, The greenbergian word order correlations, Language: Journal of the Linguistic Society of America 68 (1992) 81–138.

[9] D. M. Eberhard, G. F. Simons, C. D. Fennig, Ethnologue: Languages of the World, 28th ed., SIL International, Dallas, Texas, 2025. URL: http://www.ethnologue.com.

[10] M. A. Cysouw, Quantitative methods in typology, in: R. Köhler, G. Altmann, R. G. Piotrowski (Eds.), Quantitative Linguistics: An International Handbook, De Gruyter, Berlin; New York, 2005, pp. 554–578.

[11] D. Bakker, Language sampling, in: J. J. Song (Ed.), The Oxford Handbook of Linguistic Typology, Oxford University Press, Oxford, UK, 2010, pp. 100–128. URL: https://doi.org/10.1093/oxfordhb/9780199281251.013.0007. doi:10.1093/oxfordhb/9780199281251.013.0007, online edition published on Oxford Academic, 18 Sept. 2012.

[12] N. Evans, S. C. Levinson, The myth of language universals: Language diversity and its importance for cognitive science, Behavioral and Brain Sciences 32 (2009) 429–448. URL: https://doi.org/10.1017/S0140525X0999094X. doi:10.1017/S0140525X0999094X.

[13] B. Bickel, Typology in the 21st century: Major current developments, Linguistic Typology 11 (2007) 239–251. URL: https://doi.org/10.1515/LINGTY.2007.018. doi:10.1515/LINGTY.2007.018.

[14] T. Güldemann, The Languages and Linguistics of Africa, De Gruyter Mouton, Berlin; Boston, 2018. URL: https://doi.org/10.1515/9783110421668. doi:10.1515/9783110421668.

[15] E. Sapir, Selected Writings in Language, Culture, and Personality, University of California Press, Berkeley, CA, 1949.

[16] B. L. Whorf, Language, Thought, and Reality: Se-

lected Writings of Benjamin Lee Whorf, MIT Press, Cambridge, MA, 1956.
[17] J. A. Lucy, Grammatical Categories and Cognition: A Case Study of the Linguistic Relativity Hypothesis, Cambridge University Press, 1992. URL: https://doi.org/10.1017/CBO9780511620713. doi:10.1017/CBO9780511620713.
[18] J. Rijkhoff, D. Bakker, K. Hengeveld, P. Kahrel, A method of language sampling, Studies in Language 17 (1993) 169–203. doi:10.1075/sl.17.1.07rij.
[19] J. Rijkhoff, D. Bakker, Language sampling, Linguistic Typology 2 (1998) 263–314.
[20] A. Schleicher, O jazyku litevském, zvláště na slovanský [on the lithuanian language, and specifically on slavic], Časopis Českého Museum [Journal of the Czech Museum] 27 (1853) 320–324.
[21] A. Schleicher, Die ersten spaltungen des indogermanischen urvolkes [the first splits of the proto-indo-european people], Allgemeine Monatsschrift für Wissenschaft und Literatur [Monthly Journal of Science and Literature] 3 (1853) 786–787.
[22] W. McKinney, Data Structures for Statistical Computing in Python, in: Stéfan van der Walt, Jarrod Millman (Eds.), Proceedings of the 9th Python in Science Conference, 2010, pp. 56 – 61. doi:10.25080/Majora-92bf1922-00a.
[23] The pandas development team, pandas-dev/pandas: Pandas, 2020. URL: https://doi.org/10.5281/zenodo.3509134. doi:10.5281/zenodo.3509134.
[24] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring network structure, dynamics, and function using networkx, in: G. Varoquaux, T. Vaught, J. Millman (Eds.), Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, CA USA, 2008, pp. 11–15.

A. Full Calculation of the DV for the Example in Figure 2

tree 1

family 1

$DV = 2$ (full calculation in Section 2.2)

group 1

$L = 2$ (maximum number of levels in the sibling trees of the tree rooted at group 1)
Node count:

- $N_0 = 1$ (number of nodes at level 0)
- $N_1 = 2$ (number of nodes at level 1)

Calculating the contributions:

- $C_0 = 0$ (contribution of the root level)
- $C_1 = 0 + (2-1) \times \frac{2-(1-1)}{2} = 0 + (1 \times 1) = 1$

$DV = 1$

group 2

$L = 2$ (sibling of group 1)
Node count:

- $N_0 = 1$ (number of nodes at level 0)
- $N_1 = 2$ (number of nodes at level 1)
- $N_2 = 3$ (number of nodes at level 2)

Calculating the contributions:

- $C_0 = 0$ (contribution of the root level)
- $C_1 = 0 + (2-1) \times \frac{2-(1-1)}{2} = 0 + (1 \times 1) = 1$
- $C_2 = 1 + (3-2) \times \frac{2-(2-1)}{2} = 1 + (1 \times \frac{1}{2}) = \frac{3}{2}$

$DV = \frac{1}{2}\left(1 + \frac{3}{2}\right) = \frac{5}{4} = 1.25$

group 3

$L = 1$ (maximum number of levels in the sibling trees of the tree rooted at group 3)
Node count:

- $N_0 = 1$ (number of nodes at level 0)
- $N_1 = 2$ (number of nodes at level 1)

Calculating the contributions:

- $C_0 = 0$ (contribution of the root level)
- $C_1 = 0 + (2-1) \times \frac{2-(1-1)}{2} = 0 + (1 \times 1) = 1$

$DV = 1$

group 4

$L = 1$ (sibling of group 3)
Node count:

- $N_0 = 1$ (number of nodes at level 0)
- $N_1 = 1$ (number of nodes at level 1)

It behaves like a language isolate, so we set $DV = 1$.

tree 2

family 2

$L = 3$ (sibling of family 1)
Node count:

- $N_0 = 1$ (number of nodes at level 0)
- $N_1 = 2$ (number of nodes at level 1)
- $N_2 = 3$ (number of nodes at level 2)

Calculating the contributions:

- $C_0 = 0$ (contribution of the root level)
- $C_1 = 0 + (2-1) \times \frac{3-(1-1)}{3} = 0 + (1 \times 1) = 1$
- $C_2 = 1 + (3-2) \times \frac{3-(2-1)}{3} = 1 + (1 \times \frac{1}{3}) = \frac{4}{3}$

$DV = \frac{1}{2}\left(1 + \frac{4}{3}\right) = \frac{1}{2} \times \frac{7}{3} = \frac{7}{6} = 1.167$

group 5

$L = 1$ (maximum number of levels in the sibling trees of the tree rooted at group 5)

Node count:

- $N_0 = 1$ (number of nodes at level 0)
- $N_1 = 2$ (number of nodes at level 1)

Calculating the contributions:

- $C_0 = 0$ (contribution of the root level)
- $C_1 = 0 + (2 - 1) \times \frac{1-(1-1)}{1} = 0 + (1 \times 1) = 1$

$DV = 1$

group 6

$L = 1$ (sibling of group 5)

Node count:

- $N_0 = 1$ (number of nodes at level 0)
- $N_1 = 1$ (number of nodes at level 1)

It behaves like a language isolate, so we set $DV = 1$.

B. Example of input CSV file

id	name	parent_id	place	type
Afro-Asiatic	Afro-Asiatic	-	-	family
36	Berber	Afro-Asiatic	-	group
1793	Awjila-Sokna	1063	-	group
1063	Eastern	36	-	group
1064	Siwa	1063	-	group
37	Northern	36	-	group
1704	Atlas	37	-	group
gnc	Guanche	36	Spain	language
auj	Awjilah	1793	Libya	language
swn	Sawknah	1793	Libya	language
siz	Siwi	1064	Egypt	language
cnu	Chenoua	37	Algeria	language
jbe	Judeo-Berber	1704	Israel	language
shi	Tachelhit	1704	Morocco	language
tzm	"Tamazight, Central Atlas"	1704	Morocco	language
zgh	"Tamazight, Standard Moroccan"	1704	Morocco	language

Table 1
Sample taken from the Afro-Asiatic family tree on Ethnologue.