ULTERA Database Manual

Volume 1: Contributor's Guide

Version: 1.0 (2024-11-22)

Live version available at:

https://www.overleaf.com/read/fzpmmkmqstrt

Current Maintainers:

Adam M. Krajewski

Ricardo Amaral

Leon Hsiao

Arindam Debnath

Contents

Preface

ULtrahigh TEmperature Refractory Alloys, or ULTERA, is a multi-million project under the DOE ARPA-E ULTIMATE program, developing next-generation materials for gas turbine applications. We created an alloy data infrastructure optimized for machine learning (ML) inverse design [1], [2], hosting the world's largest database of high entropy alloys (HEAs) and their experimental properties [3]. **As of June 2024, our database encompasses over 7,000 property data points from 3,000 HEAs, sourced from over 560 publications**.

The ULTERA Database comes with sophisticated data processing, curation, and aggregation tools developed over many years. These tools have proven crucial in identifying and removing 5-10% erroneous data from literature datasets, from duplicate entries affecting error estimates to composition parsing issues impacting model performance. While most tools remain proprietary, we have released some as an open-source Python package, `PyQAlloy`. This framework supports various alloy systems, from complex concentrated solutions (HEAs, MPEAs, CCAs) to traditional alloys, and extends to non-metallic systems like High Entropy Oxides (HEOs) and Nitrides (HENs). Our auxiliary tools include `nimCSO` [4] for optimal data selection in ML applications, and `nimplex` [5], which introduces a novel graph-based representation of compositional spaces for efficient HEA screening.

This manual is organized into two volumes to serve different user groups effectively. The **Contributor's Guide** provides detailed instructions for researchers and scientists who wish to submit their experimental data to the ULTERA database, ensuring data quality and consistency with our established standards. The **Maintainer's Guide** offers comprehensive documentation for database administrators and core team members responsible for data curation, validation, and infrastructure maintenance. Together, these sections ensure the continued growth and reliability of our database while maintaining the highest standards of data quality and accessibility.

1 Contributor's Guide

1.1 How to Contribute

Contributing data to the ULTERA database follows a simplified process designed to ensure data quality and consistency. The four main steps are:

1. Make your own copy (fork) of the `ULTERA-contribute` GitHub repository from: https://github.com/PhasesResearchLab/ULTERA-contribute
2. Use the provided Excel template (`template.xlsx`) to structure your data.
3. Commit and push your completed template to your forked repository.
4. Notify the ULTERA team of your contribution by opening an issue at: https://github.com/PhasesResearchLab/ULTERA-contribute/issues.

1.1.1 GitHub Repository

The first step in contributing to ULTERA requires using GitHub, a collaborative platform that efficiently tracks file changes and versions. If you're new to GitHub, you only need a free account to get started. Simply visit github.com, sign up, and verify your email address.

With an active account, you can create your own copy (fork) of our template repository ULTERA-contribute (Figure 1), which serves as your dedicated workspace for data contribution. The process requires no software installation and can be completed entirely through a web browser following the instructions in the template repository.

1.1.2 Excel Template

In your forked repository, you'll need to update the Excel template 'template.xlsx' with your data. The template you populate with your data is in the Excel spreadsheet format and can be modified in any way that doesn't change its core structure. The format was chosen because it is universally recognized and for providing a number of conveniences, such as (1) allow styling of the dataset, ranging from, e.g., customizing column widths to the automated highlighting of values that are outside of an accepted range; (2) allow the usage of functions and automatic conversions on the fly; (3) format the displayed precision to the liking of the

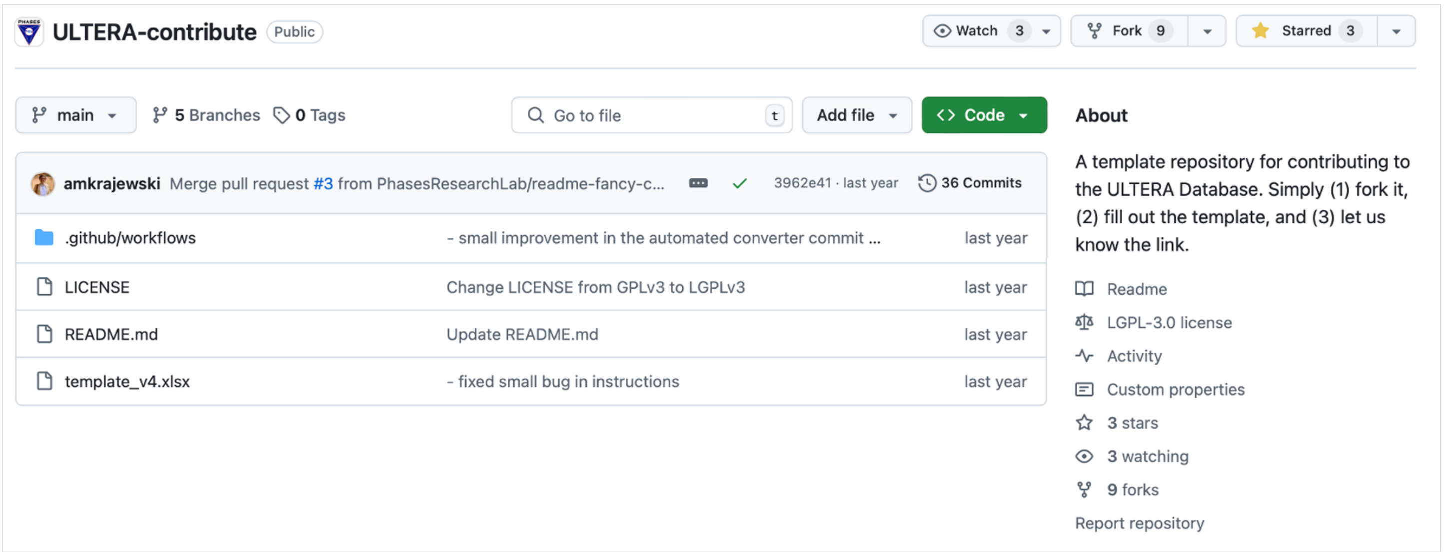

Figure 1: ULTERA-contribute template repository.

researcher or community while retaining full precision; and even (4) store extra information, such as text and images, for reference.

Contributing with new data is an interactive process that may include data points from multiple sources and contributions at different moments. To account for this, users have the flexibility to segment their data into a single or multiple Excel files, which may be organized depending on data type, time of contribution, etc. To do so, users should simply make copies of the 'template.xlsx' file in their repositories, renaming each copy to reflect the content of its data. Future contributions may occur by adding new data to the existing Excel files or making new copies of 'template.xlsx'.

Note: Avoid using only version numbers or year in the file name, as it will make correcting errors in the datasets much more difficult. Also, keep all templates in the root of your repository, i.e. not moving them inside other directories.

Figure 2 shows the Excel file 'template.xlsx' partially filled out for demonstration. The template have short descriptions and examples above each field to aid in their edition. For further information of data types see Section 1.2.

Metadata		Optional Upload Comments
Name:	FirstName LastName	Optional Comment
Email:	email@institutional.domain	

Examples:	Cr Fe Mg Ni	BCC	AC	Open field..	tensile yield strength	EXP	strain rate 1e-3, ...	298	750e6	25e6	Pa	T7 / F6 / P13	10.1557/jmr.2018.153
Optional Id	**Material**				**Property**							**Reference**	
Id/nickname	**Composition**	**Structure**	**Processing**	**Comment**	**Name**	**Source**	**Property Parameters**	**Temperature [K]**	**Value [SI]**	**Uncertainty [SI]**	**Unit [SI]**	**Pointer**	**DOI**
1	Nb45 Ta25 Ti15 Hf15	BCC	A	fabricated by ...	creep rate	EXP	stress 50 MPa	1173	1.64E-08		s-1	F2	10.1016/j.actamat.2024.119940
2	Hf0.5 Mo0.5 Nb Ta0.5 Ti VZr	BCC+HCP	H	prepared by ...	hardness	EXP	load 3 kgf	298	4.06E+09	1.06E+08	Pa	F4	10.1016/j.intermet.2024.108350
3	Al7.5 Nb30.83 Ti30.83 Zr30.83	BCC+B2	AC	produced by ..	tensile yield strength	EXP	strain rate 1e-3	295	9.75E+08	1.00E+07	Pa	T1	10.1016/j.msea.2024.146828

Figure 2: Excel template 'template.xlsx' partially filled out for demonstration.

1.1.3 Commit Changes

After updating the Excel template ‘template.xlsx‘ with your data, you’ll need to commit your changes to your fork repository. GitHub provides several methods to accomplish this, accommodating different user preferences and experience levels:

1. For inexperienced users, GitHub user interface offers a straightforward approach, allowing direct file uploads through your browser.
   i.e.: https://www.youtube.com/watch?v=ZMYs1HTRK7Q (external content).

2. For users familiar with version control systems, GitHub’s web-based editor provides an integrated environment suitable for managing multiple files and making regular updates.
   i.e.: https://www.youtube.com/watch?v=d7jHUh1PGwU (external content).

3. Alternatively, maintaining a local copy of the repository offers additional flexibility and control over the contribution process.

Besides supplying the template files for data contribution, the ULTERA-contribute repository also serves to tracking changes to your data.

To enable data tracking, an automated GitHub action will convert all of your Excel templates in the main directory into plain-text CSV files, add them to your git working branch, and, if there are any changes, commit them under ‘(automatic) Action for Data Tracking‘.

These automated actions occur every time you commit and push new contributions, and require no further action from the user.

1.1.4 Notifying your Contributions

Once you have committed your changes to your fork repository, simply notify the ULTERA team by opening an issue at: https://github.com/PhasesResearchLab/ULTERA-contribute/issues

The ULTERA team will receive a notification of your contributions and will automatically analyze and process the data into the ULTERA database.

I want to contribute in the future but I'm not ready to make it public yet

Forking a repository offers a one-click solution to clone templates, track changes, and enable public collaboration, though its mandatory public visibility means you can't keep data private. While it has a number of advantages, like enabling the community to review your data; we acknowledge that some people may want to keep their data private until they are ready to publish it.

To create a private contribution to ULTERA (or any other dataset following the template schema) you will need to import the repository. You can do so by going to the 'Create new...' in the top-right corner of GitHub page and selecting 'Import Repository'.

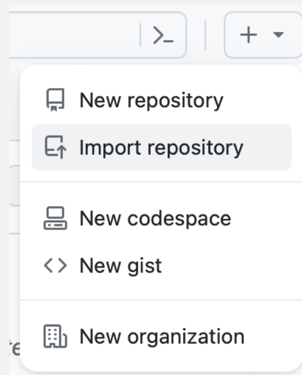

Figure 3: GitHub import screenshot.

Once the page opens, paste the URL of the original ULTERA-contribute repository:
https://github.com/PhasesResearchLab/ULTERA-contribute

Then select your repository name following the instructions in the ULTERA-contribute template repository, and select the visibility you would like to have. Go forward, wait a minute, and refresh the page; you should now see your private data repository!

Now, since it's not a fork, you can't just click a button and synchronize it, resolving all the issues on the fly in GitHub. However, if the modifications you make do not introduce any conflicts, you should be able to just add the public template repository as one of the remotes:

```
git remote add public https://github.com/PhasesResearchLab/ULTERA-contribute
```

; and then, whenever you want to make your repository up-to-date, simply pull and push changes from ULTERA-contribute to your private repo:

```
git pull public main
git push origin main
```

With that, you should be ready to have your own ULTERA-contribute repository to store all of your data privately. To contribute, simply complete the next steps as described in 1.1.2, 1.1.3, and 1.1.4.

1.2 ULTERA Input Schema

The user input data types can be classified into three broad categories: **Material** (1.2.1), **Property** (1.2.2), and **Reference** (1.2.3).

1.2.1 Material

As a descriptor for each material in consideration, the following information has to be provided by the user:

1. **Composition** - The user must input a string with alphanumeric values. Spaces are not required, but improve the visibility.

   The entered composition string must be in either one of these formats:

   (a) Fractional composition - The elements in the alloy should be reported in any order, each followed by their concentrations in fractions. **Example**: `Cr0.1 Fe0.1 Ni0.8`

   (b) Absolute composition - The elements in the alloy should be reported in any order, each followed by their absolute concentrations. **Example**: `Nb9 Hf Co3`

   (c) Equiatomic composition - For alloys with equiatomic composition, only the elements appearing in any order need to be reported. **Example**: `Cr Fe Mg Ni`

   Note: Compositions should be provided in atomic concentrations rather than weight concentrations.

2. **Structure** - The user must input all the phases occurring in the material as a single string, giving preference to the most specific and commonly used acronyms. In the case of multiple phases in the material, use '+' as a separator between each phase. No spaces should be included.

   The entered structure string must be in either one of these formats:

   - Phase acronym (capitalized): `BCC, FCC, BCT, HCP, B2, L12, L10, D03, MC, MB2, ..`
   - Phase name (non-capitalized):
     - Crystalline: `laves, sigma, gamma, martensite, ..`
     - Non-crystalline (glass, BMG, etc.): `amorphous`
   - Phase stoichiometry (chemical formula): `Ti2Ni, Al3Zr5, ..`

   Examples:

- For a single-phase alloy with BCC structure: `BCC`
- For a multiphase alloy with BCC and Laves phase: `BCC+laves`
- For a multiphase alloy with two BCC phases: `BCC+BCC`

Note: Avoid using unabbreviated names (e.g. body-centered cubic, etc), special characters (e.g. α, β, σ, etc), or general descriptive terms (e.g., single crystal, polycrystalline).

3. **Processing** - The processing conditions in abbreviated form. In case of two or more processing conditions used, the user should report them separated by '+'.

   The input provided must conform to one of the standardized abbreviations listed in Table 1. For processing techniques that are not listed, please enter the abbreviation of the parent category or, if that is inadequate, the full name of the new process.

   Example: For an alloy that has been ball milled (BM), hot isostatically pressed (HIP) and annealed (A), the entry will be: `BM+HIP+A`

4. **Comments** - Any additional comments that may be relevant to further specify the material described, such as grain size, sample thickness, time and temperature processing parameters, etc. This field can be left empty.

Table 1: Standardized processing abbreviations

Parent Category	Processing Technique
AC (As-cast)	VAM (Vacuum Arc Melting)
	AAM (Argon Arc Melting)
	VIM (Vacuum Induction Melting)
	VC (Vacuum Cast)
	DC (Die Cast)
	PC (Pressure Cast)
	IC (Investment Casting)
RS (Rapid Solidification)	MS (Melt Spinning)
PMP (Powder Metallurgical Process)	MA (Mechanical Alloying)
	BM (Ball Milling)
	SPS (Spark Plasma Sintering)
	HIP (Hot Isostatic Pressing)
AM (Additive Manufacturing)	SLM (Selective Laser Melting)
	SLS (Selective Laser Sintering)
	DED (Direct Energy Deposition)
A (Annealing)	H (Homogenization)
	RX (Recrystallization)
	N (Normalizing)
T (Tempering)	SR (Stress Relief)
Q (Quenching)	AQ/WQ/OQ (Air/Water/Oil-quench)
HW/CW (Hot/Cold Working)	SPD (Severe Plastic Deformation)
	HE/CE (Hot/Cold Extruded)
	HR/CR (Hot/Cold Rolling)
AT (Aging Treatment)	–
CT (Chemical Treatment)	ST (Solution Treatment)
	C (Carburizing)
	N (Nitriding)
	PS (Passivation)
SC (Surface Coating)	SD (Sputtering Deposition)
	TE (Thermal Evaporation)
	CVD (Chemical Vapor Deposition)
	CP (Chemical Plating)
	PC (Powder Coating)

1.2.2 Property

The property data for each material must contain the following information:

1. **Name** - Property name reported as defined in Table 2, column 'Property Name'. For properties that are not listed, please enter the full name of the new property in lowercase.

   The entered property name may or may not be preceded by a 'classifier' using one of these formats:

   (a) Estimated - Used if the property value reported is approximate or not fully determined.
   Example: `estimated tensile ductility`

   (b) True - Used if the property value is corrected to represent the intrinsic/natural material property.
   Example: `true ultimate tensile strength`

   (c) Specific - Used if the property value is normalized.
   Example: `specific creep rate`

2. **Source** - The source of data in abbreviated form on how the data has been produced.

   The entered abbreviation must follow one of these options:

   (a) **EXP** - EXPerimental

   (b) **EMP** - EMPirical Modeling

   (c) **ML** - Machine Learning

   (d) **DFT** - Density Functional Theory

   (e) **CAL** - CALPHAD Modeling

3. **Property Parameters** - Any additional parameter required to fully define a property value, such as stress-strain loading rate. See examples in the Observations column of Table 2. This field may be left empty.

4. **Temperature** - The temperature, *in Kelvin*, at which the data was reported. Must be a numerical value.

5. **Value** - The property value reported in S.I. units. The entry must be a numeric value.
   Example: For the compressive yield stress of a material reported as 5 megapascals (MPa), the user should report it as 5×10^6 or `5000000` pascals.

6. **Uncertainty** - The property value uncertainty reported in S.I. units. The entry must be a numeric value.

7. **Unit** - The S.I. property value unit as a string.

Table 2: Standardized property names

Property Name	**Thesaurus**	**Observations**
density	density = ρ	-
melting temperature solidus temperature liquidus temperature	melting temperature = Tm	Specify pressure conditions if not at ambient pressure.
tensile yield strength compressive yield strength bending yield strength	yield strength = YS strength = stress bending = flexural	Stress measured at 0.2% strain deformation. Specify loading rate.
ultimate tensile strength ultimate compressive strength ultimate bending strength	ultimate tensile strength = UTS strength = stress bending = flexural	Stress measured at maximum of stress-strain curve. Specify loading rate.
fracture tensile strength fracture compressive strength fracture bending strength	strength = stress bending = flexural	Stress measured at fracture. Specify loading rate.
tensile ductility compressive ductility bending ductility	ductility = elongation = strain = ϵ bending = flexural	Strain measured at failure. If not defined, estimated at maximum strain measured. Specify loading rate.
youngs modulus	youngs modulus = elastic modulus youngs modulus = Y = E	Strain/stress relation measured under uniaxial stress.
bulk modulus	bulk modulus = K = B	Strain/stress relation measured under volumetric stress.
shear modulus	shear modulus = G	Strain/stress relation measured under shear stress.
B/G ratio	-	Ratio between bulk and shear modulus.
poissons ratio	poissons ratio = ν	Ratio between transverse and axial strain under uniaxial loading.
hardness	hardness = microhardness	Specify measurement method (HV, HB, HR, etc.) and load conditions.
nanohardness	-	Specify maximum load.
fracture toughness	fracture toughness = KIC	Measured under mode I loading.
thermal expansion coefficient	thermal expansion coefficient = CTE thermal expansion coefficient = α	Specify temperature range.
thermal conductivity	thermal conductivity = heat conductivity thermal conductivity = κ = λ	Specify temperature range.
creep rate	creep rate = minimum creep rate	Specify stress condition of measurement.

1.2.3 Reference

1. **Reference Pointer** - The position of data in the reference paper. The entry *must be* coded in either of these formats
   - **T** - Table, followed by table number
   - **F** - Figure, followed by figure number
   - **A** - Appendix, followed by appendix section
   - **P** - Page, followed by page number. (Only if no other reference is available in the paper)
   - **S** - Supplementary Materials, followed by one of the 4 above, e.g. `SF2, ST4`
2. **Reference DOI** - The user must provide a unique identifier, i.e. DOI, for the paper from which they have extracted the data. In cases the DOI cannot be found, the users must use the abbreviation scheme: First author's last name (FA), year (YR) and three *first* words from title (3WFT) formatted as "FA_YR_3WFT" to report the paper used.

   Example: For *O.N. Senkov et al.: Development and exploration of refractory high entropy alloys—A review*, the reference *has to be* the DOI `10.1557/jmr.2018.153`; or, in case the paper was from before-DOI-times: `Senkov_2018_DevelopmentAndExploration`

Table 3 summarizes all the user input data types adopted in ULTERA.

Table 3: Summary of ULTERA Input Schema

	Field	Description	Data Type	Entry Format
Material	Composition	The elements in the alloy along with their respective concentrations	Alphanumeric String	A[A] B[B] ..
	Structure	The phases present in the alloy	Alphanumeric String	Phase1+Phase2+ ..
	Processing	The processing conditions in abbreviated form	Alphanumeric String	Process1+Process2+..
	Comments	Additional comments from the user regarding the material	Alphanumeric String	Open field
Property	Name	Property name of interest	String	Standard property name/ Full property name
	Source	How the data was generated	String	EXP/EMP/ML/DFT/CAL
	Parameters	Additional information required to fully define a property value	String	Open field
	Temperature	The temperature at which property was reported, in Kelvins	Numeric	e.g. 298, 1000, ..
	Value	The measured property in SI units	Numeric	e.g. 5×10^6, 5000000, ..
	Uncertainty	The property value uncertainty in SI units	Numeric	e.g. 1×10^6, 1000000, ..
	Unit	The S.I. property value unit	Alphanumeric String	e.g. Pa, m3, kg, s-1, ..
Reference	Pointer	The position of data in the reference paper	Alphanumeric String	e.g. F7, T2, P3, SF2, ..
	DOI	The paper from which the data was taken	Alphanumeric String	DOI/FA_YR_3WFT

References

[1] A. Debnath, A. M. Krajewski, H. Sun, S. Lin, M. Ahn, W. Li, S. Priya, J. Singh, S. Shang, A. M. Beese, Z.-K. Liu, and W. F. Reinhart, “Generative deep learning as a tool for inverse design of high entropy refractory alloys,” *Journal of Materials Informatics*, vol. 1, no. 1, p. 3, Sep. 2021. DOI: 10.20517/jmi.2021.05. [Online]. Available: https://www.oaepublish.com/articles/jmi.2021.05.

[2] A. Debnath, L. Raman, W. Li, A. M. Krajewski, M. Ahn, S. Lin, S. Shang, A. M. Beese, Z.-K. Liu, and W. F. Reinhart, “Comparing forward and inverse design paradigms: A case study on refractory high-entropy alloys,” *Journal of Materials Research*, vol. 38, no. 17, pp. 4107–4117, Sep. 2023. DOI: 10.1557/s43578-023-01122-6. [Online]. Available: https://link.springer.com/10.1557/s43578-023-01122-6.

[3] W. Li, L. Raman, A. Debnath, M. Ahn, S. Lin, A. M. Krajewski, S. Shang, S. Priya, W. F. Reinhart, Z.-K. Liu, and A. M. Beese, “Design and validation of refractory alloys using machine learning, CALPHAD, and experiments,” *International Journal of Refractory Metals and Hard Materials*, vol. 121, p. 106 673, Jun. 2024. DOI: 10.1016/j.ijrmhm.2024.106673. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0263436824001215.

[4] A. M. Krajewski, A. Debnath, W. F. Reinhart, A. M. Beese, and Z.-K. Liu, “nimCSO: A Nim package for Compositional Space Optimization,” Mar. 2024. DOI: 10.48550/arXiv.2403.02340. [Online]. Available: https://arxiv.org/abs/2403.02340v1.

[5] A. M. Krajewski, A. M. Beese, W. F. Reinhart, and Z.-K. Liu, “Efficient Generation of Grids and Traversal Graphs in Compositional Spaces towards Exploration and Path Planning Exemplified in Materials,” Feb. 2024. DOI: 10.48550/arXiv.2402.03528. [Online]. Available: http://arxiv.org/abs/2402.03528.