Integrating EMalign into ChimeraX:

A Bundle for Enhanced Volume Alignment in Structural Biology

**1. Abstract**

This paper describes an extension for ChimeraX to enable the direct alignment of two density maps using the EMalign algorithm (Harpaz & Shkolnisky, 2023). EMalign aligns three-dimensional density maps by exploiting common lines between projection images, automatically handling rotations, reflections, and translations. Integration of EMalign into ChimeraX provides a user-friendly alignment tool, enhancing the functionality of ChimeraX and simplifying the process for structural biologists.

**2. Introduction**

In the realm of computational structural biology, accurate alignment of density maps is indispensable for the analysis and comparison of molecular structures. Precise alignment enables researchers to understand structural variations and interactions at a molecular level, which is critical for advancements in fields such as drug discovery and protein engineering. However, traditional alignment methods often involve multiple, labor-intensive steps that require specialized tools. These methods can be time-consuming and prone to errors, highlighting the need for more efficient solutions.

ChimeraX (Meng EC, 2023), a next-generation molecular visualization program offers advanced visualization and analysis capabilities. Its user-friendly interface and robust features make it a widely used tool in various research settings. Yet, despite its strengths, the challenge of efficient volume alignment persists.

To address this, an extension integrating the EMalign algorithm into ChimeraX was developed. EMalign leverages common lines between projection images to align three-dimensional density maps. It handles rotations, reflections, and translations automatically, providing a sophisticated solution to the alignment problem.

By embedding EMalign into ChimeraX, an intuitive and user-friendly method for the precise alignment of density maps is offered. This integration not only enhances the functionality of ChimeraX but also significantly improves the overall user experience in structural biology research, making complex alignment tasks more accessible and efficient.

**3. Methods**

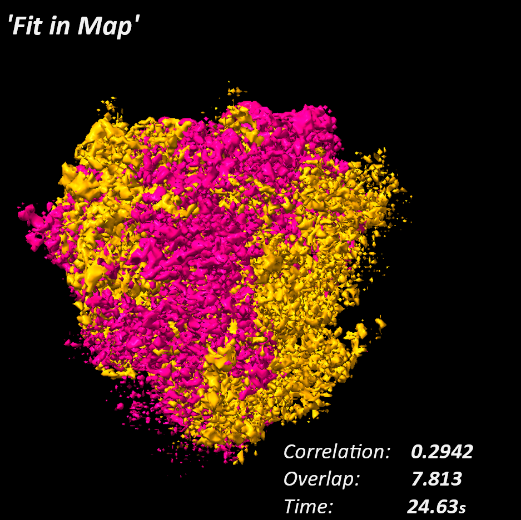
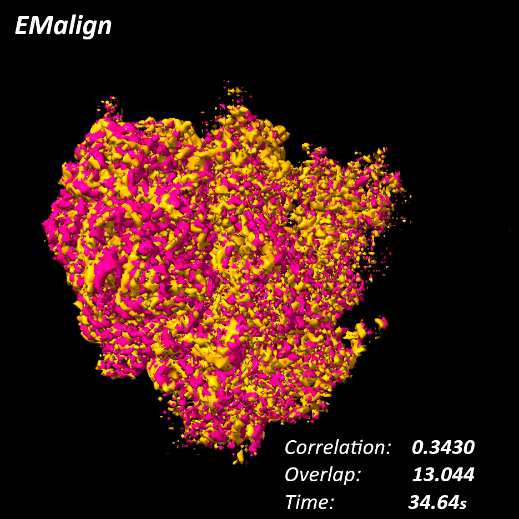
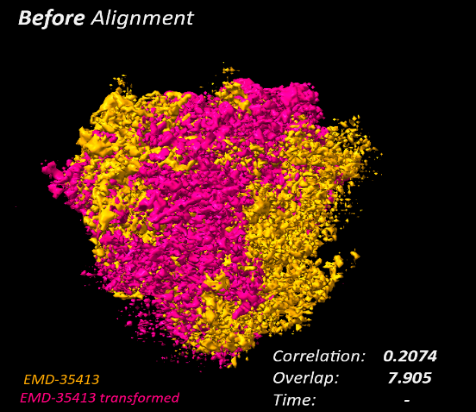
**3.1. 'Fit in Map' – ChimeraX's Built-In Alignment Tool**

'Fit in Map' – this volume alignment tool, integrated within the ChimeraX platform, optimizes the placement and alignment of molecular structures within density maps. This tool employs a rigid-body local optimization algorithm, which iteratively refines the position of the fit model to maximize the overlap or correlation with the reference map. The optimization process involves calculating the optimal rotational and translational adjustments and continues until either convergence is achieved or a predefined number of iterations is reached.

Prior to calling 'Fit in Map', the user has to perform a preliminary manual alignment. This is important since the local optimization algorithm initiates adjustments from the current position of the model. By manually positioning the model close to the desired alignment, the user ensures that the algorithm can effectively fine-tune the fit from a reasonable starting point. Without this initial rough alignment, the tool might not achieve the best possible fit, especially if the initial position is significantly far from the correct alignment.

**Figure 1** The top figure displays the map EMD-35413 alongside a randomly transformed version of it, prior to any alignment.

The figures below present two independent alignments: on the left, the alignment achieved using EMalign, and on the right, the alignment obtained using the 'Fit in Map' tool (executed without an initial manual alignment).



**3.2. Overview of the EMalign Algorithm**

The **EMalign algorithm** aligns two three-dimensional density maps, denoted as φ₁ and φ₂, which are related by a rotation matrix and a three-dimensional translation. The process involves the following steps:

* **Projection Generation**: Multiple projection images (P₁, P₂, ..., Pₙ) are generated from φ₂ at different orientations (R₁, R₂, ..., Rₙ).
* **Alignment**: Each projection image is aligned with φ₁ to estimate the rotation so that projecting φ₁ in the orientation determined by will result in the image . Thus providing estimations for each .
* **Optimization**: An optimization problem is solved to estimate the transformation matrix O that aligns φ₂ with φ₁. The key equation used is

where is the Frobenius matrix norm. This optimization ensures that the sum of the squared differences between the estimated transformations and the actual rotations is minimized.

* **Handling Transformations**: The algorithm automatically handles rotations, reflections (handedness), and translations between the maps, making it fully automatic and robust.
* **Final Alignment**: The optimal transformation matrix is applied to φ₂, aligning it with φ₁ for accurate comparisons and analyses.

The core idea of EMalign is to leverage common lines between projection images to achieve high-precision alignment of density maps.

**3.3. Implementation**

3.3.1. Integration with ChimeraX:

The integration of EMalign into ChimeraX involved several key steps:

* **Familiarization with ChimeraX**: The process started by becoming highly familiar with ChimeraX's functionalities, understanding the data representation of maps and the purpose of each tool in the 'Volume' menu to ensure that the EMalign tool would fit within this context.
* **Review of ChimeraX**: A thorough examination of its source code helped in grasping the programming methodologies used within ChimeraX. It was crucial for effectively integrating EMalign, ensuring compatibility and seamless operation within ChimeraX. Additionally, by becoming familiar with the full functionalities of ChimeraX, the review ensured that existing functions and commands could be utilized, avoiding unnecessary work.

3.3.2. Modifications and enhancements to the EMalign algorithm:

**A. Essential modifications for functionality within ChimeraX**

1. **Interface creation:** An interface was developed to integrate ChimeraX's source code objects and functions with the EMalign algorithm. The original algorithm utilized ndarrays for volume representations, whereas ChimeraX employs a unique 'Model' object containing fields such as origin, step, and cell angles. Raw data were extracted from the Volume objects and converted into a 3D array format compatible with the EMalign algorithm.
2. **User-friendly graphical tool:** A graphical tool was designed to enhance the usability of the EMalign algorithm within ChimeraX. This tool allows users to execute the alignment by either manually entering the new EMalign command in ChimeraX's command line or by utilizing the newly added EMalign tool in the 'Volume' menu. This addition simplifies the process, making it more accessible and user-friendly. Furthermore, the tool provides more options for specific parameters of the alignment, whereas the manual command facilitates only basic alignment. Previously, users had to manually install and execute EMalign from a terminal before importing the aligned maps into ChimeraX for further analysis. Comprehensive instructions for utilizing this tool are provided in the 'Graphic Tool Guide' section under 'Implementation'.
3. **Replacing pyFFTW with SciPy:** During the initial run of the original EMalign package post-installation via the pip install command, an issue was encountered with the package pyFFTW, a dependency required by EMalign. pyFFTW is a Python wrapper for FFTW 3, a high-speed Fast Fourier Transform library, intended to offer a comprehensive interface for all possible transforms executed by FFTW. It aligns with the NumPy and SciPy FFT interfaces to optimize their performance by leveraging FFTW's speed with minimal code modifications. Despite no errors during the installation, attempts to execute EMalign in the terminal resulted in recurring errors importing DLL files. To resolve this, all instances of pyFFTW were replaced with the SciPy library, which provides robust functionality for Fast Fourier Transform (FFT) operations. Although this change caused a slight increase in running time, it ensured compatibility and maintained the accuracy of the alignment algorithm.

**B. Modifications made to EMalign to improve performances:**

These modifications were necessitated by the outcomes observed during the testing conducted for Test Case A (see section 4.3), which aimed to verify that the EMalign algorithm yields consistent results when executed both in the terminal and within the ChimeraX extension. While results were consistent between both versions, some maps exhibited a lack of the desired robustness. For instance, map EMD-35413 produced 8 out of 20 alignments with correlations inferior to those of the original unaligned maps, indicating a need for enhanced robustness and reliability within ChimeraX.

The observed inconsistency in results for the same map across various random seeds suggested that the issue was related to the random sections of the code. The original EMalign algorithm performs several steps to align the volumes. It takes the reference volume, the query volume, and other optional variables such as (number of projections to use) and (downsample size) as inputs. Initially, 15,236 rotation matrices are generated and stored in a variable called . From this set, rotation matrices are randomly selected and applied to the query volume to generate projection images. Subsequently, the same process is repeated for the reference volume by randomly selecting and using rotation matrices to generate projection images.

The modifications involved consolidating the random selection of rotation matrices into a single step, thereby allowing control over which matrices were used for each volume. Initially, the first sampled matrices were applied to the query volume, and the latter to the reference volume.

* 1. **Switching the assignment of the rotation matrices**: After obtaining a correlation value for the initial direction of assignment (query to reference), the algorithm also applied the first matrices to the reference volume and the latter to the query volume. This reverse order was evaluated, and the direction yielding the higher correlation value for the downsampled volumes was chosen for the final alignment parameters.
  2. **Re-Selecting the rotation matrices:** Upon the alignment of the downsampled volumes as described above, two new random sets of rotation matrices are sampled. The alignment parameters are subsequently re-evaluated employing this new selection. The parameters that yield the highest correlation for the downsampled volumes are then selected and applied to the original volumes.

**C. Modification to the optimization process:**

**Replacing the BFGS optimization**: The original BFGS optimization step was replaced with ChimeraX's built-in 'Fit in Map' command, which performs similar final refinements.

BFGS Algorithm – The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is a widely used iterative method for solving unconstrained nonlinear optimization problems. Named after its developers, this algorithm is particularly valued for its efficiency and robustness. It approximates the second-order Hessian matrix of the objective function by iteratively updating an inverse Hessian matrix using gradient evaluations. The BFGS method is known for its superlinear convergence properties, making it a powerful tool for optimizing complex functions with a high degree of accuracy. Within the context of EMalign, the BFGS algorithm refines the alignment of density maps by optimizing their correlation values, thereby enhancing alignment quality and ensuring minimal residual errors.

'Fit in Map' – this volume alignment tool, integrated within the ChimeraX platform, optimizes the placement and alignment of molecular structures within density maps. Utilizing a gradient based search algorithm, 'Fit in Map' identifies the optimal fit of a model within an electron microscopy (EM) map, considering both rotational and translational adjustments to ensure accurate positioning within the density map. The 'Fit in Map' tool capitalizes on ChimeraX's robust optimization algorithms, which handle large datasets and complex molecular structures efficiently. In the context of the EMalign algorithm, the 'Fit in Map' feature is employed to further optimize the alignment of maps aligned by EMalign, thereby enhancing the accuracy and quality of density maps’ alignments.

The 'Fit in Map' command alone is insufficient for optimal alignment of volumes, as it necessitates an initial manual alignment by the user to achieve the best possible overlap prior to fine-tuning. As a result, the alignment accuracy is heavily dependent on the quality of the user's manual alignment. The EMalign algorithm automates the entire alignment process, obviating the need for any pre-processing or manual alignment, which constitutes a significant advantage. However, even with EMalign’s automatic alignment, the application of 'Fit in Map' for additional refinement can significantly increase the precision of the final result. Therefore, an optional checkbox (default: checked) was incorporated into the EMalign tool's options menu, allowing users to determine whether to execute 'Fit in Map' at the conclusion of the EMalign alignment. If selected, the command is automatically executed, thereby providing a more refined alignment.Consequently, the necessity for the BFGS algorithm was obviated, leading to its removal to enhance computational efficiency and reduce running times.

**D. Modifications made to add new functionalities to EMalign:**

* 1. **Alignment of maps with different dimensions**: During beta testing with biologists who frequently use ChimeraX, the need for aligning maps of different sizes was highlighted. Similar volumes often arise due to biological reasons, such as different states of a molecule or varied conformations of a protein. The dimensions of the maps can differ because they imaged under different conditions, even though the underlying structures are similar. Functionality was added to align density maps with different dimensions. For example, if the reference map's grid size () is smaller and its voxel size () is larger than the query map's grid size () and voxel size (), the following steps are taken:
* Calculate the size: .
* Downsample the query map to the size: and crop it to the size .
* Perform the alignment to obtain rotation (), translation (), and reflection parameters. The parameter is a 3D matrix representing the optimal rotation calculated for the query map, while is a vector representing the optimal translation. Adjust to be correctly applied on the original size map: .
* Apply the rotation and the adjusted translation to the original query map for the final alignment.
  1. **Masking:** Masking involves creating a "mask" to define a specific region of interest within a density map or volume based on certain spatial criteria. This mask functions as a filter, including the desired region and excluding irrelevant areas. By excluding noisy or irrelevant regions, masking enhances the quality of the reconstructed region, thereby contributing to a more accurate alignment. We thus added to EMalign a technique known as soft masking. Soft masks are characterized by gradual transitions between included and excluded regions, achieved through smoothing or thresholding to avoid sharp boundaries. The region of interest was defined by calculating the radius of a sphere that encompassed 90% of the volume's energy.

3.3.3. Utilizing existing ChimeraX bundles:

The integration of the EMalign tool within ChimeraX required the use of several key libraries from ChimeraX's source code, known as bundles. These provided the necessary functionality for the integration and enhancement of the alignment capabilities. The relevant bundles used in the project include:

* **Core**: Provides fundamental functionalities and utilities, mainly used to create the new command and the corresponding tool. This usage is more technical and not specific to the EMalign algorithm.
* **Map**: Handles the objects of the maps as they are given by ChimeraX, which are then converted into the format used originally by EMalign.
* **MapData**: Manages the data structures for maps.
* **MapFit**: Contains the 'Fit in Map' tool, which was used for the optimization of the alignment as a final refinement step.
* **UI**: Used to create the graphics of the EMalign tool, including the options menu, a list of all open maps to choose for the reference map and the query map, and other user interface elements.

3.3.4. Conversion and compatibility:

ChimeraX represents maps with specialized objects that contain multiple fields encompassing all pertinent data. By thoroughly familiarizing with ChimeraX's source code, particularly the bundles, it became feasible to accurately extract and convert the required data into the format suitable for EMalign. This process involved retrieving raw data from the Volume objects and transforming it into a compatible 3D array format (ndarray).

3.3.5. Command and tool creation:

The process of creating the new command and graphical tool for EMalign followed the comprehensive guide provided on the ChimeraX website for adding a bundle. The steps included:

1. **Adding a command**:
   * **Create bundle information**: Start by creating a bundle\_info.xml file containing information about the bundle, such as name, version, and dependencies.
   * **Implement command functionality**: Develop a Python package that interfaces with ChimeraX and implements the command functionality. This typically involves creating a Python file (e.g., cmd.py) that defines the command.
   * **Build bundle**: Package the bundle as a Python wheel. This can be done using tools like make or directly through ChimeraX.
   * **Install and test**: Install the bundle in ChimeraX using the toolshed install command and test its functionality.
2. **Adding a tool**:
   * **Create bundle information**: Similar to adding a command, start by creating a bundle\_info.xml file.
   * **Implement tool functionality**: Develop a Python package that interfaces with ChimeraX and implements the tool’s functionality. This involves creating a Python file (e.g., tool.py) that defines the tool.
   * **Build bundle**: Package the bundle as a Python wheel.

**Install and test**: Install the bundle in ChimeraX using the toolshed install command and test its functionality.

3.3.6. The EMalign command via the ChimeraX command line:

To run EMalign without the graphic tool, the user can use one of the following calls from the command line in ChimeraX:

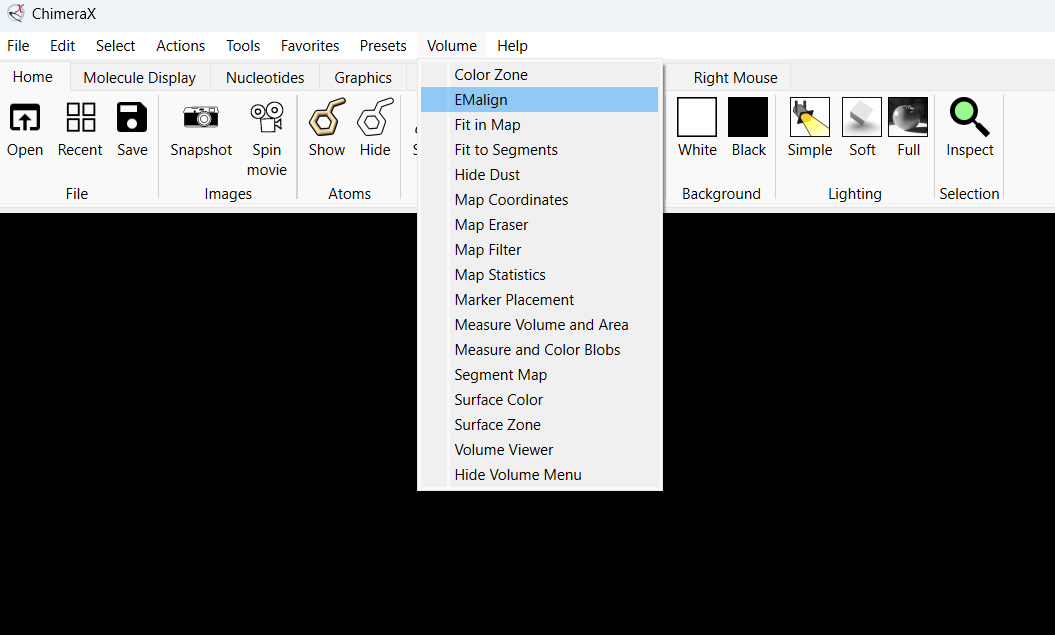
[volume emalign](help:user/commands/volume.html#emalign) #1 queryMap #2

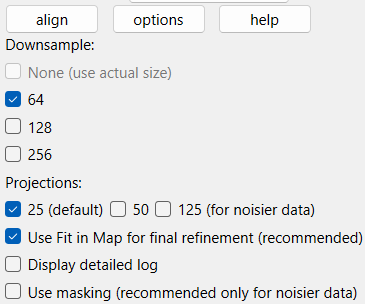
[vol emalign](help:user/commands/volume.html#emalign) #1 queryMap #2

* #1 – the Model ID of the reference map
* #2 – the Model ID of the query map

3.3.7. Graphic tool guide:

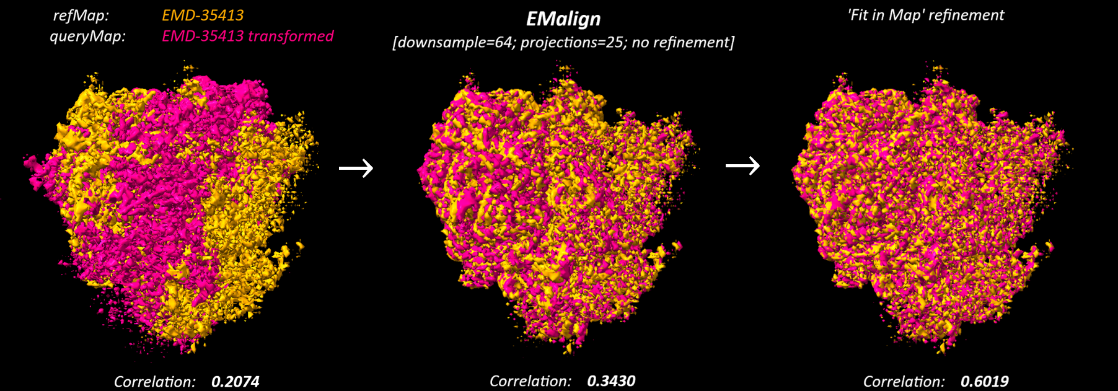
To enhance the usability of the EMalign algorithm within ChimeraX, a graphical tool was created. This section provides a short guide on how to use this tool, with screen captures to illustrate each step.

1. **Opening the EMalign tool**: Open ChimeraX and navigate to the 'Volume' menu. Select the EMalign tool from the list.
2. A screenshot of a computer

   Description automatically generated**Selecting maps**: In the EMalign tool window, select the reference map and the query map from the dropdown menus.
3. **Setting alignment parameters**: Set the parameters for alignment, such as rotation, translation, and reflection options. Adjust these settings as needed for your specific use case.

1. A screenshot of a computer

   Description automatically generated**Running the alignment**: Click the 'Align' button to start the alignment process. The tool will automatically handle the alignment and provide a progress log when finished.

1. **Viewing the aligned maps**: Once the alignment is complete, you can view the aligned maps directly in ChimeraX. The aligned maps will be displayed, and you can further analyze or manipulate them as needed.

**4. Testing methodologies**

4.1.Test environment:

* **Software**: The tests were conducted both within the ChimeraX program and using the PyCharm integrated development environment (IDE).

The tests employed ChimeraX version 1.8, Python version 3.11, and the EMalign algorithm version 1.0.5 throughout the testing process.

* **Operating System**: Windows 11 for ChimeraX and Python, and Linux (Ubuntu 20.04) for ChimeraX to ensure cross-platform compatibility.

4.2. Test data:

**Maps**: All maps used in this section were obtained from the Electron Microscopy Data Bank (EMDB). The following maps were selected for their diversity in structure and biological relevance:

* **EMD-2660:** Cryo-EM structure of the Plasmodium falciparum 80S ribosome bound to the anti-protozoan drug emetine. Map size [360, 360, 360].
* **EMD-19195:** Cryo-EM structure of the rabbit 80S ribosome stalled on a 2-TMD rhodopsin intermediate in complex with Sec61-RAMP4. Map size [420, 420, 420].
* **EMD-35413:** Cryo-EM structure of the human 80S ribosome in complex with dibekacin. Map size [640, 640, 640][[1]](#footnote-1).

For each reference map above, a query map was generated by applying a set of random transformations (rotations, reflections, and translations).

**Test rounds**: To ensure the elimination of random variability and enhance the reliability of the results, distinct random seeds were employed for each round. The seed number for each round is identical across all versions of the EMalign algorithm. A list of unique seed numbers was sampled randomly once, with each seed corresponding to a distinct round of testing.

4.3.Test cases:

**Introduction:** To evaluate the performance and accuracy of the EMalign algorithm within the ChimeraX platform, several test cases have been designed corresponding to the modifications made to the original EMalign package. For a detailed description of these modifications, refer to section 3.3.2, "Modifications and enhancements to the EMalign algorithm".

Test case A – Essential functional consistency:

* **Objective**: To verify that the EMalign algorithm yields consistent alignment results when integrated within the ChimeraX platform, as compared to its standalone execution from the terminal. This test case is executed on the preliminary version of EMalign, which incorporates only the fundamental modifications necessary to enable operational functionality within the ChimeraX environment.
* **Alignment procedure**: For each map, rounds of alignment were conducted for the query map to the reference map in two scenarios:
* Once using the original EMalign from the terminal.
* Once using the modified EMalign running in ChimeraX.

Test case B – Performance improvements:

* **Objective**: To assess the enhancements in alignment performance and robustness of the EMalign algorithm, following specific modifications applied exclusively to the version integrated within the ChimeraX platform. An additional test was designed to evaluate computational times for each version of EMalign.
* **Alignment procedure**: For each map, rounds of alignment were conducted for the query map to the reference map using the EMalign algorithm, modified as detailed in section 3.3.2, sub-section B.
* **Computational time evaluation**: Six downsized maps were created from the original map EMD-35413, which has an initial size of 640 voxels. The downsized maps were generated with the following sizes: 160, 240, 320, 400, 480, and 560 voxels. Each version of the EMalign algorithm was subsequently executed on all six downsized maps.

Test case C – BFGS algorithm vs. 'Fit in Map':

* **Objective**: To conduct a comparative analysis of the optimization methods employed by both the original EMalign algorithm and its ChimeraX-integrated counterpart. Specifically, this study aims to evaluate the performance and accuracy of the BFGS algorithm utilized in the original EMalign, and the 'Fit in Map' feature implemented within the ChimeraX platform for volume alignment optimization.
* **Procedure**:
* **Maps**: The maps used are those that have been aligned using the original versions of EMalign in Test Case A, for each seed number. These maps are post-alignment but pre-optimization.
* **Algorithm Comparison**:
  1. **BFGS Algorithm**: Utilized in the original EMalign for further optimization of aligned volumes.
  2. **Fit in Map**: Utilized in the ChimeraX-integrated EMalign for optimization, leveraging the existing ChimeraX feature.
* **Optimization Procedure**: For each post-alignment map from Test Case A, optimization were applied directly using both the BFGS algorithm and the 'Fit in Map' feature.

**5. Results**

5.1. Essential functional consistency (Test case A):

**Objective:** The aim of this test case was to demonstrate that when making only the essential modifications required for the EMalign algorithm to function within the ChimeraX platform, the performance remains comparable to the original EMalign package executed from the terminal.

**Results summary:** The table below summarizes the average correlation values for each map before alignment, after alignment with the original EMalign package executed from the terminal, and after alignment with the ChimeraX-integrated version. It also includes the difference between the latter two columns.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMD id** | **size** | **pre-**  **alignment** | **original module** | **ChimeraX module** | **difference** |
| 2660 | 360 | 0.0341 | 0.131475 | 0.13146 | 0.000015 |
| 19195 | 420 | 0.2821 | 0.383115 | 0.382995 | 0.00012 |
| 35413 | 320 | 0.2072 | 0.245245 | 0.246815 | 0.00157 |

**Interpretation:** The results indicate that the essential functionality of the EMalign algorithm remains consistent and reliable when integrated within the ChimeraX platform, as evidenced by the minimal absolute differences in correlation values between the original terminal execution and the ChimeraX-integrated version.

**Complete results:** The detailed results for each map, including individual test values, are provided in the appendix at the end of this document.

5.2. Performance improvements (Test case B):

**Objective:** The objective of this test case was to compare the performances of the difference modifications made to the EMalign module to establish that these changes contributed to the overall quality and improved robustness of the algorithm. For a detailed description of these modifications see section 3.3.2.

**Description of the EMalign versions tested:**

* **Original:** The initial version of EMalign, which contains only the necessary modifications to function, as described in section 3.3.2, subsection A.
* **B1:** This version also tests the switching of the rotation matrices when creating projections, as described in section 3.3.2, subsection B.1.
* **B2:** This version tests the re-run of the original version with a new random selection of rotation matrices. The re-run is limited to a single repetition, as described in section 3.3.2, subsection B.2.
* **B (B1 + B2):** This version is the combination of versions B1 and B2. This means that we run B1 with the initial random selection, then run B1 again with a new random selection of rotation matrices.

**Results[[2]](#footnote-2):**

The table below includes the average correlation values for the original version and the three modified versions (B1, B2, B). It provides an overview of the performance improvements across different versions of the EMalign algorithm.

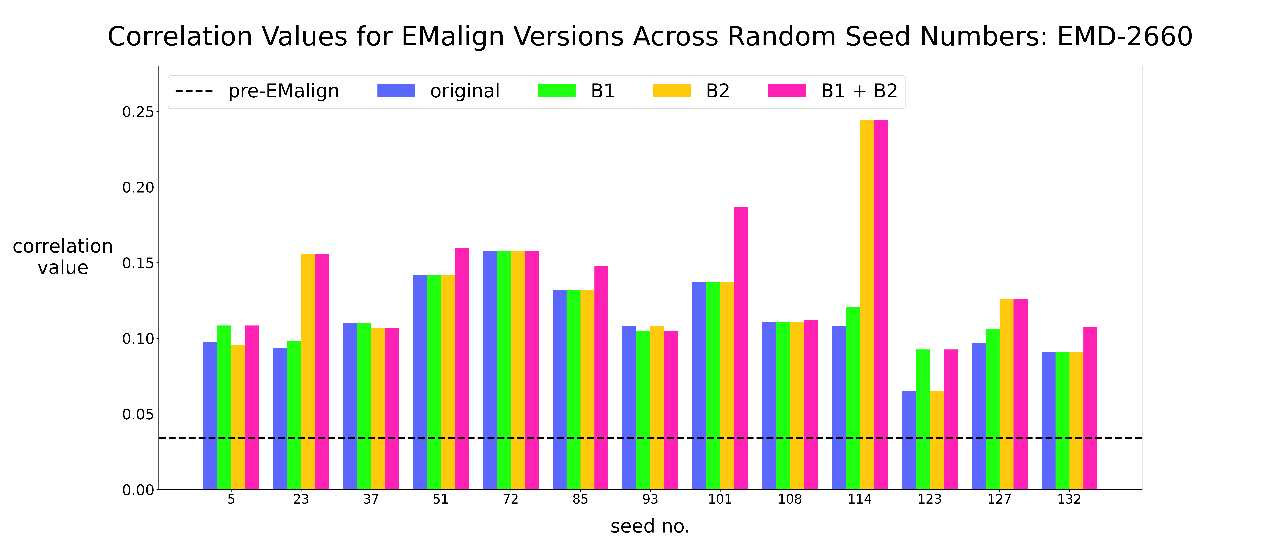
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EMD id** | **size** | **pre-alignment** | **original EMalign** | **B1** | **B2** | **B (B1+B2)** |
| 2660 | 360 | 0.0341 | 0.1315 | 0.1345 | 0.1426 | 0.1495 |
| 19195 | 420 | 0.2821 | 0.3830 | 0.3911 | 0.3934 | 0.4005 |
| 35413 | 320 | 0.2072 | 0.2468 | 0.2960 | 0.2724 | 0.3272 |

Each individual modification (B1 and B2) shows an increase in the correlation values compared to the original version, indicating that these modifications positively impacted the algorithm's alignment accuracy and robustness. The combination of both modifications (B) demonstrates the most significant improvement, suggesting that the integration of both strategies enhances the overall performance of the algorithm.

EMD-2660

Upon detailed examination of the full results, it is evident that the majority of the test rounds (11 out of 20) exhibited an insignificant correlation difference of less than 0.005 between the highest and the lowest results. Of these, seven seeds showed no difference at all and were therefore excluded from the plot. Among the remaining rounds, most (7 out of 9) displayed a slightly higher difference in correlation values, yet still relatively insignificant, with a correlation difference of less than 0.05.

Notably, two test rounds demonstrated significant improvements, with a correlation difference greater than 0.05 between at least two of the four EMalign versions tested. Specifically, for seed number 23, versions B2 and B exhibited an improvement of more than 150% compared to either B1 or the original version. Similarly, for seed number 114, the highest improvement was observed, with over 200% enhancement between versions B2 or B and B1 or the original.



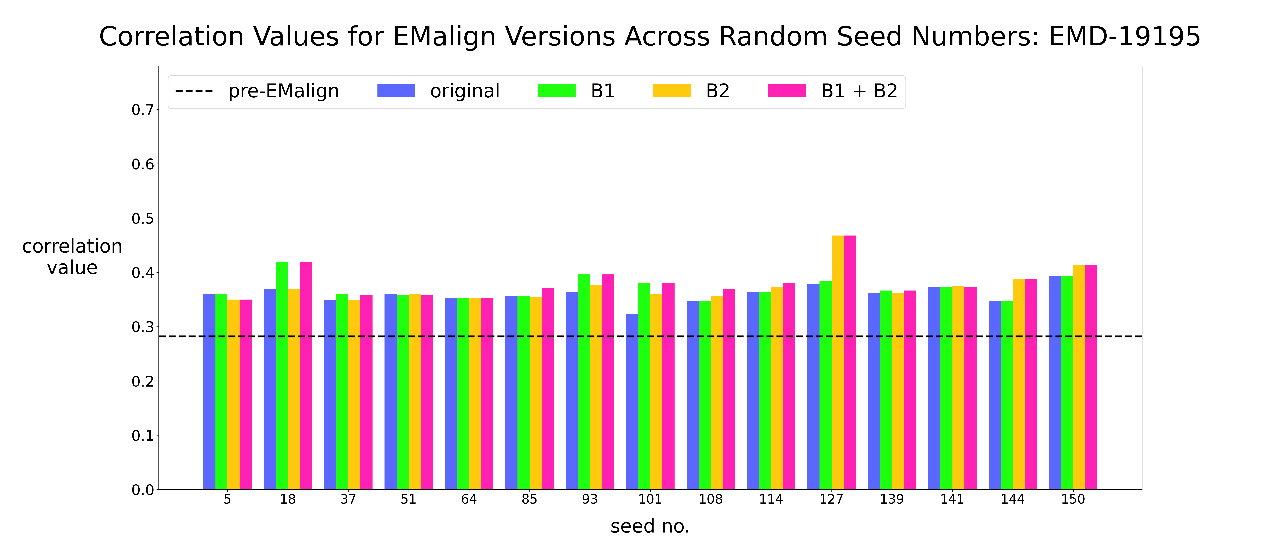
As illustrated in the plot above, for seeds 5 and 123, versions B1 and B achieved equal correlation values, whereas version B2 demonstrated poorer performance. Conversely, for seeds 23, 114, and 127, versions B2 and B achieved the highest correlation values.

Moreover, for seeds 51, 85, 101, 108, and 132 version B achieved the highest correlation values exclusively. However, seeds 37 and 93 yielded the worst correlation results for version B. The latter seems unusual, since version B incorporates both versions B1 and B2 and should therefore perform at least as well as the better of the two versions.

EMD-19195

Contrary to the previous map, a smaller number of test rounds (9 out of 20) for this map exhibited an insignificant correlation difference of less than 0.005. Of these, five seeds yielded equal results across all versions and were therefore excluded from the plot.

Among the remaining rounds, none displayed a correlation difference exceeding 0.09, with the majority (9 out of 11) exhibiting differences below 0.05. These results demonstrate that both B1 and B, as well as B2 and B, have their respective strengths across different seed numbers.



Similarly to the map EMD-2660, several seeds (5, 37, 141) did not result in version B having the highest correlation value. For seed 5, this difference is illustrated in the plot above, while for the two other seeds, the difference is significantly lower and is only noticeable in the detailed results.

It is also worth noting the seemingly balanced performance observed between version B1 and version B2. In three seeds (18, 93, 101), versions B1 and B achieved higher correlation values, while in another three seeds (127, 144, 150), versions B2 and B exhibited superior performance.

Overall, it is difficult to ascertain which version, between B1 and B2, performs best based on the plot above, as both demonstrated similarly inconsistent performances. Detailed results indicate that version B2 holds a slight advantage over version B1, with average correlation values of 0.3934 and 0.3911, respectively. However, version B still exhibited the best overall performance, yielding an average correlation value of 0.4005.

EMD-35413

Following an in-depth review of the results, several key observations have been identified. Two seeds (37 and 101) produced equal values across all four EMalign versions tested, and thus were excluded from the plot as they did not provide any insight into the effect of the modifications of subsection B on the algorithm.

A graph of different colored lines

Description automatically generated with medium confidence

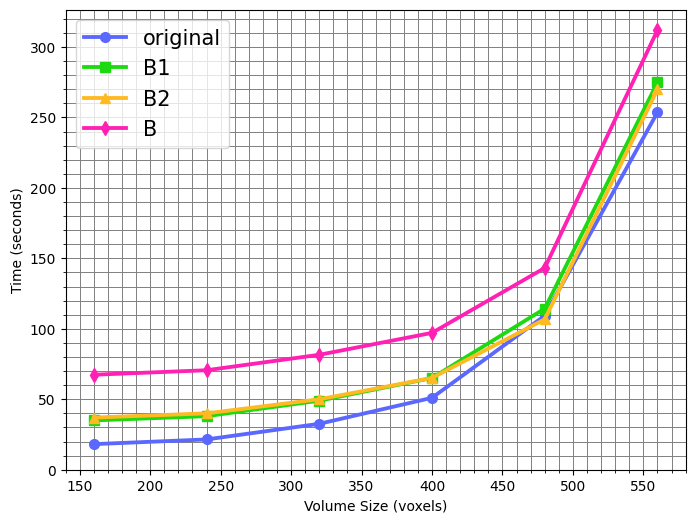
As can be seen in the plot above, seven seeds resulted with correlation values worse than the initial (unaligned) volumes, for at least one version of EMalign. Specifically, seeds 23 and 123 yielded results with diminished correlation only for the original EMalign version. Seeds 114 and 132 demonstrated reduced correlation for both the original EMalign and B1 versions, while seeds 51, 141, and 150 exhibited worse correlation than the initial volumes for both the original EMalign and B2 versions.

Interestingly, seed 49 was unique in that version B yielded the worst results among the four EMalign versions tested, while the remaining three versions presented equally higher results.

Upon examining the rounds for which only a single version achieved the highest correlation values, it was found that seven seeds (5, 72, 93, 127, 139, 141, and 150) presented the highest correlation solely for version B, despite it being a combination of versions B1 and B2. Further analysis of rounds where exactly two versions received equal highest results revealed that nine seeds (18, 23, 51, 64, 85, 108, 114, 123, and 132) exhibited the highest correlation for version B alongside either version B1 or B2, but not both. Specifically, five seeds (18, 51, 64, 85, and 108) demonstrated the best results for versions B and B1, whereas four seeds (23, 114, 123, and 132) showed the best results for versions B and B2.

Computational time evaluation

The plot below demonstrates the computational run-times as a function of map size. The original algorithm exhibited the lowest computational time, whereas version B showed the highest. Versions B1 and B2 exhibited nearly identical run-times across all volume sizes. Starting from a size of 400 voxels, the performance of the original, B1, and B2 versions converges. Nevertheless, the difference in run-time between versions B1 or B2 and version B remains consistently at approximately 30 seconds, regardless of the map size.



**Conclusion:**

While there were a few instances where version B did not achieve the highest result among all versions, the difference between the highest score and that attained by version B in these instances was rather insignificant. Overall, version B consistently achieved the highest correlation values, underscoring its robustness and overall effectiveness. In many instances, version B was the sole version to achieve the highest correlation, suggesting that it employs a unique assignment of rotation matrices that is not attempted by either version B1 or B2 alone. Conversely, the data indicated specific seeds where, in addition to version B, either version B1 or version B2 excelled, but not both.

This variability implies that certain modifications may be more effective under specific conditions, further emphasizing the significance of version B, which encompasses the strengths of both versions B1 and B2. Nevertheless, it is important to acknowledge that version B, despite its robustness and effectiveness, is the most time-consuming among all versions. This increased computational time may present a challenge for users who prioritize speed over accuracy.

5.3. Optimization comparison – BFGS algorithm vs. 'Fit in Map' (Test case C):

**Objective:** This test case aims to perform a comparative analysis of the optimization methods employed by both the original EMalign algorithm and its ChimeraX-integrated counterpart. The original version of the latter is used to ensure the results truly reflect the quality of the optimization algorithms. Specifically, this study evaluates the optimization and accuracy of the BFGS algorithm utilized in the original EMalign, and the 'Fit in Map' feature implemented within the ChimeraX platform for volume alignment optimization.

The results for each map are presented separately. A summarized table includes the correlation of the maps pre-alignment, the average correlation values for the original version and the optimized versions using both the BFGS algorithm and the 'Fit in Map' feature. An interpretation of the results is provided for each map to offer a comprehensive understanding of the optimization differences. Additionally, plots are provided to visualize the results for each map.

**Results Summary[[3]](#footnote-3):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMD id** | **size** | **pre-alignment** | **BFGS** | **'Fit in Map'** | **difference** |
| 2660 | 360 | 0.0341 | 0.9113 | 0.9565 | 0.0452 |
| 19195 | 420 | 0.2821 | 0.6554 | 0.9412 | 0.2858 |
| 35413 | 320 | 0.2072 | 0.5316 | 0.7110 | 0.1794 |

**Interpretation**: The summary table provides an overview of the optimization differences between both methods. The comparison includes the BFGS algorithm and the 'Fit in Map' feature. Each method's optimization is evaluated based on the correlation values, highlighting the effectiveness of the optimization techniques. The "difference" column indicates the improvements achieved by using the 'Fit in Map' feature compared to the BFGS algorithm. The higher correlation values and significant differences demonstrate the superior performance and accuracy of the 'Fit in Map' feature.

Next, the results for each map will be presented using plots to provide a visual representation of the data.

EMD-2660

The optimization results for EMD-2660 indicate that both the BFGS algorithm and the 'Fit in Map' feature achieve high correlation values close to 1.0 across all seed numbers, with some variation between the two algorithms. The 'Fit in Map' feature achieves a slightly higher average correlation value (0.9565) compared to BFGS (0.9113), with a difference of 0.0452. However, this difference is not substantial enough to conclusively determine that 'Fit in Map' is superior to the BFGS algorithm for this dataset. The results suggest that both methods are effective for aligning this particular volume.

A yellow and purple lines

Description automatically generated

EMD-19195

The optimization results for EMD-19195 indicate a notable difference in performance between the BFGS algorithm and the 'Fit in Map' feature. The 'Fit in Map' feature consistently achieves higher correlation values compared to the BFGS algorithm across different seed numbers. The average correlation value for 'Fit in Map' is 0.9412, whereas the BFGS algorithm achieves 0.6554, resulting in a substantial difference of 0.2858. This suggests that the 'Fit in Map' feature provides a more accurate alignment for this dataset, demonstrating its enhanced effectiveness in optimizing the alignment of density maps. Furthermore, this significant improvement supports the interpretation from the previous map (EMD-2660), where the difference in correlation values was not sufficient on its own to conclusively determine the superiority of the 'Fit in Map' feature.

A graph with yellow and purple lines

Description automatically generated

EMD-35413

The overall results for EMD-35413 initially appear comparable to those of the previous map, exhibiting an average correlation of 0.5316 for the BFGS algorithm and 0.7110 for the 'Fit in Map' feature. However, a detailed analysis reveals a more nuanced performance trend. By segmenting the results into two distinct groups and analyzing them independently, a significant pattern is evident.

A graph with yellow and purple lines

Description automatically generated

For half of the rounds (seeds: 18, 37, 49, 64, 72, 85, 101, 127, 139, 144), the results indicate that both the BFGS algorithm and the 'Fit in Map' feature are relatively equally effective, with average correlations of 0.8446 and 0.9015, respectively. This suggests that in these scenarios, both methods are effective, with 'Fit in Map' demonstrating a slight advantage.

In contrast, the other half of the rounds display a different trend. Both algorithms exhibit reduced performance; however, the BFGS algorithm performed significantly worse, with an average correlation of 0.2186, compared to 0.5205 for the 'Fit in Map' feature. Notably, in 5 out of these 10 rounds, the alignment process using the BFGS algorithm actually decreased the correlation between the maps to such an extent that the resulting values were lower than the initial correlation between the unaligned maps. This finding underscores a critical weakness in the BFGS algorithm's performance under specific conditions. Although the 'Fit in Map' results were also lower in these rounds compared to others, all results still indicate an improvement in correlation, albeit minimal.

This substantial variation further corroborates the interpretation from the previous map (EMD-19195), where a notable difference in correlation values was also observed. The consistent superior performance of the 'Fit in Map' feature across all tests confirms its effectiveness and reliability in optimizing the alignment of density maps.

Considering that 'Fit in Map' is employed post-alignment on whichever EMalign version will eventually be integrated into ChimeraX, it is prudent to test this feature on maps aligned with the default version of the ChimeraX-integrated EMalign (version B2). Since the only problematic map appears to be EMD-35413, this additional test will be limited to it.

A graph of numbers and lines

Description automatically generated with medium confidence

As illustrated in the plot above, focusing on seeds 23, 51, 93, 114, and 150, there is a marked improvement (near 1.0 correlation) when applying 'Fit in Map' to the map aligned with version B2 of EMalign for ChimeraX. Notably, these are also the same 5 seeds that demonstrated poorer alignments than the unaligned maps, even after optimization with the BFGS algorithm.

**6. Discussion**

Summary of key findings

We demonstrated that integrating the EMalign algorithm into ChimeraX allows the user to perform high-quality alignment of density maps, without requiring a rough manual alignment first. The modified versions of EMalign provide more robust and accurate alignments compared to the original version of EMalign. Additionally, the integration into ChimeraX rendered EMalign easily accessible and user-friendly.

Limitations

As demonstrated by Test case A, there can occasionally be a suboptimal random selection of rotation matrices, which may further misalign the maps. This issue necessitated the development of EMalign versions B1 and B2, which aim to mitigate this risk. However, these modifications also resulted in increased computational time, which can be substantial, particularly for larger-scale volumes.

Test case B also revealed some unusual results, with specific seeds where version B did not achieve the highest correlation values, despite encompassing all three versions (original, B1, B2). A plausible explanation for this anomaly stems from the downsampling process. The correlations used to select the optimal rotation matrices are computed on the downsampled volumes. Consequently, it is possible that the leading correlation, chosen by a marginal difference, may become more significant when the corresponding alignment parameters are applied to the original volumes. Unfortunately, due to the substantial computational time required for aligning the full-sized volumes, performing the alignment multiple times is not feasible, especially for larger-scale volumes.

Solutions

To minimize the possibility of producing an alignment worse than the unaligned volumes without significantly increasing run times, two potential solutions can be implemented. One approach involves using the size of the original volumes as a factor for determining which version to employ. The larger the volume size, the longer the time required to align the full-size volumes after acquiring the alignment parameters. Therefore, above a certain volume size, versions B1 or B2 would be employed, whereas below that threshold, version B would be utilized. Alternatively, users could be provided with the option to prioritize either speed or accuracy. If speed is prioritized, versions B1 or B2 can be employed. Conversely, if accuracy is the priority, version B will be utilized.

**7. References**

\* need to add references \*

**8. Appendix:**

Table A [section 5.1]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| seed no. | **EMD-2660** | | | | **EMD-19195** | | | | **EMD-35413** | | | |
| pre-EMalign | **original module** | **ChimeraX module** | diff. | pre-EMalign | **original module** | **ChimeraX module** | diff. | pre-EMalign | **original module** | **ChimeraX version** | diff. |
| 5 | 0.0341 | 0.0973 | 0.0973 | 0 | 0.2821 | 0.3603 | 0.3603 | 0 | 0.2072 | 0.2765 | 0.2785 | 0.002 |
| 18 | 0.1682 | 0.1682 | 0 | 0.3688 | 0.3688 | 0 | 0.2508 | 0.2508 | 0 |
| 23 | 0.0931 | 0.0935 | 0.0004 | 0.3941 | 0.3941 | 0 | 0.1538 | 0.1538 | 0 |
| 37 | 0.1098 | 0.1098 | 0 | 0.3487 | 0.3487 | 0 | 0.3444 | 0.3444 | 0 |
| 49 | 0.1165 | 0.1165 | 0 | 0.3644 | 0.3644 | 0 | 0.3201 | 0.3201 | 0 |
| 51 | 0.1415 | 0.1415 | 0 | 0.3598 | 0.3598 | 0 | 0.1785 | 0.1785 | 0 |
| 64 | 0.1485 | 0.1485 | 0 | 0.3516 | 0.3519 | 0.0003 | 0.3546 | 0.3546 | 0 |
| 72 | 0.1576 | 0.1576 | 0 | 0.3807 | 0.3807 | 0 | 0.2784 | 0.2784 | 0 |
| 85 | 0.1317 | 0.1317 | 0 | 0.3568 | 0.3568 | 0 | 0.2712 | 0.2712 | 0 |
| 93 | 0.1083 | 0.1083 | 0 | 0.3652 | 0.363 | 0.0022 | 0.1804 | 0.2098 | 0.0294 |
| 101 | 0.1382 | 0.1372 | 0.001 | 0.3247 | 0.3238 | 0.0009 | 0.278 | 0.278 | 0 |
| 108 | 0.1106 | 0.1106 | 0 | 0.3478 | 0.3478 | 0 | 0.2517 | 0.2517 | 0 |
| 114 | 0.1083 | 0.1083 | 0 | 0.3633 | 0.3637 | 0.0004 | 0.2009 | 0.2009 | 0 |
| 123 | 0.0653 | 0.0653 | 0 | 0.3882 | 0.3882 | 0 | 0.1964 | 0.1964 | 0 |
| 127 | 0.0971 | 0.0971 | 0 | 0.3788 | 0.3788 | 0 | 0.2972 | 0.2972 | 0 |
| 132 | 0.0908 | 0.0908 | 0 | 0.7344 | 0.7344 | 0 | 0.1926 | 0.1926 | 0 |
| 139 | 0.209 | 0.2093 | 0.0003 | 0.3612 | 0.3612 | 0 | 0.2745 | 0.2745 | 0 |
| 141 | 0.1231 | 0.1231 | 0 | 0.3724 | 0.3724 | 0 | 0.1891 | 0.1891 | 0 |
| 144 | 0.2215 | 0.2215 | 0 | 0.3473 | 0.3473 | 0 | 0.2465 | 0.2465 | 0 |
| 150 | 0.1931 | 0.1931 | 0 | 0.3938 | 0.3938 | 0 | 0.1693 | 0.1693 | 0 |
| **avg.** |  | **0.13147** | **0.13146** |  |  | **0.38311** | **0.38299** |  |  | **0.24524** | **0.24681** |  |

Table B/EMD-2660 [section 5.2]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| seed no. | pre-EMalign | **original** | **B1** | **B2** | **B** |
| 5 | 0.0341 | 0.0973 | 0.1084 | 0.0957 | 0.1084 |
| 18 | 0.1682 | 0.1682 | 0.1682 | 0.1682 |
| 23 | 0.0935 | 0.0982 | 0.1554 | 0.1554 |
| 37 | 0.1098 | 0.1098 | 0.107 | 0.107 |
| 49 | 0.1165 | 0.1165 | 0.1165 | 0.1165 |
| 51 | 0.1415 | 0.1415 | 0.1415 | 0.1599 |
| 64 | 0.1485 | 0.1485 | 0.1485 | 0.1485 |
| 72 | 0.1576 | 0.1576 | 0.1577 | 0.1577 |
| 85 | 0.1317 | 0.1317 | 0.1317 | 0.148 |
| 93 | 0.1083 | 0.1051 | 0.1083 | 0.1051 |
| 101 | 0.1372 | 0.1372 | 0.1372 | 0.1867 |
| 108 | 0.1106 | 0.1106 | 0.1106 | 0.1119 |
| 114 | 0.1083 | 0.1208 | 0.2441 | 0.2441 |
| 123 | 0.0653 | 0.0926 | 0.0653 | 0.0926 |
| 127 | 0.0971 | 0.1059 | 0.1261 | 0.1261 |
| 132 | 0.0908 | 0.0908 | 0.0908 | 0.1076 |
| 139 | 0.2093 | 0.2093 | 0.2093 | 0.2093 |
| 141 | 0.1231 | 0.1231 | 0.1231 | 0.1231 |
| 144 | 0.2215 | 0.2215 | 0.2215 | 0.2215 |
| 150 | 0.1931 | 0.1931 | 0.1931 | 0.1931 |
| **avg.** |  | **0.1315** | **0.1345** | **0.1426** | **0.1495** |

Plot B/EMD-2660 [section 5.2]:

A graph of different colored lines

Description automatically generated

Table B/EMD-19195 [section 5.2]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| seed no. | pre-EMalign | **original** | **B1** | **B2** | **B** |
| 5 | 0.2821 | 0.3603 | 0.3603 | 0.35 | 0.35 |
| 18 | 0.3688 | 0.419 | 0.3688 | 0.419 |
| 23 | 0.3941 | 0.3941 | 0.3941 | 0.3941 |
| 37 | 0.3487 | 0.3595 | 0.3487 | 0.359 |
| 49 | 0.3644 | 0.3644 | 0.3644 | 0.3644 |
| 51 | 0.3598 | 0.358 | 0.3598 | 0.358 |
| 64 | 0.3519 | 0.3519 | 0.3533 | 0.3533 |
| 72 | 0.3807 | 0.3807 | 0.3807 | 0.3807 |
| 85 | 0.3568 | 0.3568 | 0.355 | 0.3713 |
| 93 | 0.363 | 0.3961 | 0.3774 | 0.3961 |
| 101 | 0.3238 | 0.381 | 0.3595 | 0.381 |
| 108 | 0.3478 | 0.3478 | 0.356 | 0.3693 |
| 114 | 0.3637 | 0.3637 | 0.3722 | 0.381 |
| 123 | 0.3882 | 0.3882 | 0.3882 | 0.3882 |
| 127 | 0.3788 | 0.3841 | 0.4678 | 0.4678 |
| 132 | 0.7344 | 0.7344 | 0.7344 | 0.7344 |
| 139 | 0.3612 | 0.3665 | 0.3612 | 0.3665 |
| 141 | 0.3724 | 0.3734 | 0.3739 | 0.3734 |
| 144 | 0.3473 | 0.3473 | 0.3882 | 0.3882 |
| 150 | 0.3938 | 0.3938 | 0.4137 | 0.4137 |
| **avg.** |  | **0.383** | **0.3911** | **0.3934** | **0.4005** |

Plot B/EMD-19195 [section 5.2]:

A graph of different colored lines

Description automatically generated

Table B/EMD-35413 [section 5.2]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| seed no. | pre-EMalign | **original** | **B1** | **B2** | **B** |
| 5 | 0.2072 | 0.2785 | 0.2785 | 0.2785 | 0.3561 |
| 18 | 0.2508 | 0.2749 | 0.2508 | 0.2749 |
| 23 | 0.1538 | 0.2861 | 0.2875 | 0.2875 |
| 37 | 0.3444 | 0.3444 | 0.3444 | 0.3444 |
| 49 | 0.3201 | 0.3201 | 0.3201 | 0.2927 |
| 51 | 0.1785 | 0.2887 | 0.162 | 0.2887 |
| 64 | 0.3546 | 0.4499 | 0.3546 | 0.4499 |
| 72 | 0.2784 | 0.2784 | 0.2784 | 0.3101 |
| 85 | 0.2712 | 0.5281 | 0.3108 | 0.5281 |
| 93 | 0.2098 | 0.271 | 0.2138 | 0.3227 |
| 101 | 0.278 | 0.278 | 0.278 | 0.278 |
| 108 | 0.2517 | 0.323 | 0.2517 | 0.323 |
| 114 | 0.2009 | 0.2009 | 0.2747 | 0.2747 |
| 123 | 0.1964 | 0.3029 | 0.3992 | 0.3992 |
| 127 | 0.2972 | 0.2972 | 0.2972 | 0.3295 |
| 132 | 0.1926 | 0.1926 | 0.2708 | 0.2708 |
| 139 | 0.2745 | 0.2745 | 0.2745 | 0.3213 |
| 141 | 0.1891 | 0.2484 | 0.1891 | 0.2962 |
| 144 | 0.2465 | 0.2465 | 0.2701 | 0.3085 |
| 150 | 0.1693 | 0.2364 | 0.1419 | 0.2874 |
| **avg.** |  | **0.2468** | **0.296** | **0.2724** | **0.3272** |

Plot B/EMD-35413 [section 5.2]:

A graph of different colored lines

Description automatically generated

Table C/EMD-2660 [section 5.3]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| seed no. | pre-EMalign | **BFGS** | **Fit in Map** | corr. diff. |
| 5 | 0.0341 | 0.9334 | 0.9564 | 0.023 |
| 18 | 0.9176 | 0.9566 | 0.039 |
| 23 | 0.8429 | 0.9563 | 0.1134 |
| 37 | 0.9317 | 0.9564 | 0.0247 |
| 49 | 0.922 | 0.9567 | 0.0347 |
| 51 | 0.9207 | 0.9566 | 0.0359 |
| 64 | 0.9277 | 0.9566 | 0.0289 |
| 72 | 0.9005 | 0.9572 | 0.0567 |
| 85 | 0.925 | 0.9565 | 0.0315 |
| 93 | 0.93 | 0.9559 | 0.0259 |
| 101 | 0.9167 | 0.9571 | 0.0404 |
| 108 | 0.9131 | 0.9565 | 0.0434 |
| 114 | 0.865 | 0.9557 | 0.0907 |
| 123 | 0.9316 | 0.9564 | 0.0248 |
| 127 | 0.9249 | 0.9562 | 0.0313 |
| 132 | 0.8815 | 0.9557 | 0.0742 |
| 139 | 0.9143 | 0.9569 | 0.0426 |
| 141 | 0.9183 | 0.9559 | 0.0376 |
| 144 | 0.885 | 0.9572 | 0.0722 |
| 150 | 0.924 | 0.9564 | 0.0324 |
| **avg.** |  | **0.9113** | **0.9565** |  |

Table C/ EMD-19195 [section 5.3]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| seed no. | pre-EMalign | **BFGS** | **Fit in Map** | corr. diff. |
| 5 | 0.2821 | 0.5734 | 0.9422 | 0.3688 |
| 18 | 0.7425 | 0.9408 | 0.1983 |
| 23 | 0.6452 | 0.9415 | 0.2963 |
| 37 | 0.6654 | 0.9418 | 0.2764 |
| 49 | 0.5543 | 0.9409 | 0.3866 |
| 51 | 0.8342 | 0.9417 | 0.1075 |
| 64 | 0.6575 | 0.9417 | 0.2842 |
| 72 | 0.5984 | 0.9413 | 0.3429 |
| 85 | 0.6086 | 0.9411 | 0.3325 |
| 93 | 0.7474 | 0.9406 | 0.1932 |
| 101 | 0.3352 | 0.9385 | 0.6033 |
| 108 | 0.6031 | 0.9406 | 0.3375 |
| 114 | 0.7346 | 0.9412 | 0.2066 |
| 123 | 0.8003 | 0.9415 | 0.1412 |
| 127 | 0.5647 | 0.941 | 0.3763 |
| 132 | 0.7344 | 0.9403 | 0.2059 |
| 139 | 0.5988 | 0.9416 | 0.3428 |
| 141 | 0.8004 | 0.9415 | 0.1411 |
| 144 | 0.7065 | 0.9417 | 0.2352 |
| 150 | 0.603 | 0.9415 | 0.3385 |
| **avg.** |  | **0.6554** | **0.9412** |  |

Table C/ EMD-35413 [section 5.3]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| seed no. | pre-EMalign | **BFGS** | **Fit in Map** | corr. diff. |
| 5 | 0.2072 | 0.3488 | 0.8995 | 0.5507 |
| 18 | 0.8512 | 0.8938 | 0.0426 |
| 23 | 0.1538 | 0.2544 | 0.1006 |
| 37 | 0.868 | 0.9059 | 0.0379 |
| 49 | 0.8048 | 0.904 | 0.0992 |
| 51 | 0.2019 | 0.2576 | 0.0557 |
| 64 | 0.8344 | 0.9053 | 0.0709 |
| 72 | 0.8542 | 0.9033 | 0.0491 |
| 85 | 0.8514 | 0.905 | 0.0536 |
| 93 | 0.182 | 0.3006 | 0.1186 |
| 101 | 0.8565 | 0.9029 | 0.0464 |
| 108 | 0.2808 | 0.8964 | 0.6156 |
| 114 | 0.2189 | 0.3026 | 0.0837 |
| 123 | 0.2138 | 0.2927 | 0.0789 |
| 127 | 0.8584 | 0.9018 | 0.0434 |
| 132 | 0.1881 | 0.8714 | 0.6833 |
| 139 | 0.8121 | 0.9007 | 0.0886 |
| 141 | 0.2286 | 0.8776 | 0.649 |
| 144 | 0.8554 | 0.8926 | 0.0372 |
| 150 | 0.1693 | 0.2526 | 0.0833 |
| **avg.** |  | **0.5316** | **0.7110** |  |

***TO DO:***

* Add a test showing the quality of 'Fit in Map' as the primary alignment tool, versus EMalign.

Note: since it requires a manual initial alignment it might be hard to assess.

* Add tests demonstrating the modification mentioned in subsection D under section 3.3.2 (different-size maps).
* A~~dd a plot demonstrating for each version of modified EMalign the running time as a function of map size.~~

***TBD:***

* Determine whether we want to leave the decision to the user on whether to prioritize speed or accuracy, or if we want to decide it ourselves. Build both versions so it's available to use when decided.
* Figure out whether to remove the T function from the volumes in emalign\_cmd.py – that was put in there to match the tests done so far but it seems to be unnecessary, though harmless.
* Default projections number – I think we should change to 25.

1. A downsized version of this map was utilized. Downsampled map size [320, 320, 320]. [↑](#footnote-ref-1)
2. Individual test values and complete plots are available in the appendix. [↑](#footnote-ref-2)
3. Complete results, including individual test values, are provided in the appendix. [↑](#footnote-ref-3)