JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

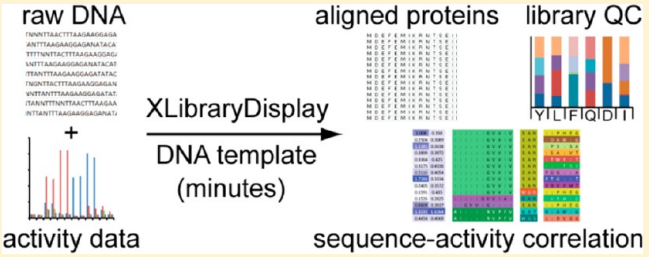
pubs.acs.org/jcim

A General Sequence Processing and Analysis Program for Protein Engineering

Ryan L. Stafford,* Erik S. Zimmerman, Trevor J. Hallam, and Aaron K. Sato

Protein Engineering, Sutro Biopharma, Inc., 310 Utah Avenue Suite 150, South San Francisco, California 94080, United States

ABSTRACT: Protein engineering projects often amass numerous raw DNA sequences, but no readily available software combines sequence processing and activity correlation required for efficient lead identification. XLibraryDisplay is an open source program integrated into Microsoft Excel for Windows that automates batch sequence processing via a simple step-by-step, menu-driven graphical user interface. XLibraryDisplay accepts any DNA template which is used as a basis for trimming, filtering, translating, and aligning hundreds to thousands of sequences (raw, FASTA, or Phred PHD file formats). Key steps for library characterization through lead discovery are available including library composition analysis, filtering by experimental data, graphing and correlating to experimental data, alignment to structural data extracted from PDB files, and generation of PyMOL visualization scripts. Though larger data sets can be handled, the program is best suited for analyzing approximately 10 000 or fewer leads or naïve clones which have been characterized using Sanger sequencing and other experimental approaches. XLibraryDisplay can be downloaded for free from sourceforge.net/projects/xlibrarydisplay/.



INTRODUCTION

Projects involving the discovery of novel proteins with new or improved functions usually accumulate large numbers of DNA sequences requiring many analysis steps. At a minimum, the raw sequences must be filtered and the open reading frames located, translated, and aligned. Then unique sequences also need to be identified and oftentimes correlated to functional data. Countless tools exist to perform DNA and protein sequence analysis but are generally fragmented and cannot efficiently process the volume of sequencing data for most protein engineering discovery efforts. For instance, many great bioinformatics resources are freely available on Internet sites such as ExPASy[1] or the Sequence Manipulation Suite (which can also be run locally).[2] In principle, it is possible to analyze each individual sequence from a phage or ribosome display selection through separate web portals and then use spreadsheet software like Microsoft Excel to relate sequences to other experimental data. This can become extremely time-consuming for more than a handful of sequences so integrated tools are preferable.

Many free and commercial software packages can be installed locally like Sequencher, Vector NTI, Geneious, BioEdit,[3] BioWord,[4] or Jalview[5] which integrate many high-quality sequence analysis tools, but they have also not been optimized for analyzing sequence and activity data from selections. Thus, many biotechnology companies often create their own customized tools to assist with sequence analysis from selections like SGCOUNT (Genentech/Roche),[6] blaze2 (Cambridge Antibody Technology/MedImmune/AstraZeneca),[7] SeqAgent/XAbTracker (Xoma),[8] or repurposed databases intended for small-molecule drug discovery (Xencor).[9] Recently, the program SARVisionlBiologics was developed which offers a number of tools for protein sequence-activity analysis, but raw DNA sequences must be processed first before proteins can be correlated to experimental data.[10] This is similar to PFAAT which provides a number of sophisticated tools for analyzing sequence-activity relationships once raw DNA sequences have been processed, translated, and aligned.[11] Usually, new programs are written for the specific goals of each discovery project, like SGCOUNT which is intended for counting amino acids in combinatorial alanine or homolog scanning libraries. Thus, there is a need for customizable sequence analysis tools that are tailored for individual protein selection and screening campaigns. Python, R, and Java are a few languages scientists frequently employ for such projects due to the availability of general-purpose bioinformatics libraries like Biopython,[12] Bioconductor,[13] and Biojava.[14]

XLibraryDisplay is a free, open source application integrated into Microsoft Excel for Windows that enables rapid analysis of libraries of DNA and protein sequences for discovery projects. The program allows data sets up to roughly 10 000 sequences to be analyzed on a single personal computer in a matter of minutes. The program exploits the fact that many libraries are derived from a common DNA scaffold from which novel proteins are selected or screened. For example, phage and ribosome display enable selections for proteins which bind targets from naïve libraries based on a single antibody framework,[15] transcription factor,[16] or other protein scaffold with favorable biophysical characteristics, e.g. DARPins[17] or fibronectin type III domains.[18] The affinity maturation of lead antibodies[19] or the directed evolution of enzymes[20] is also

Received: June 19, 2014

   dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

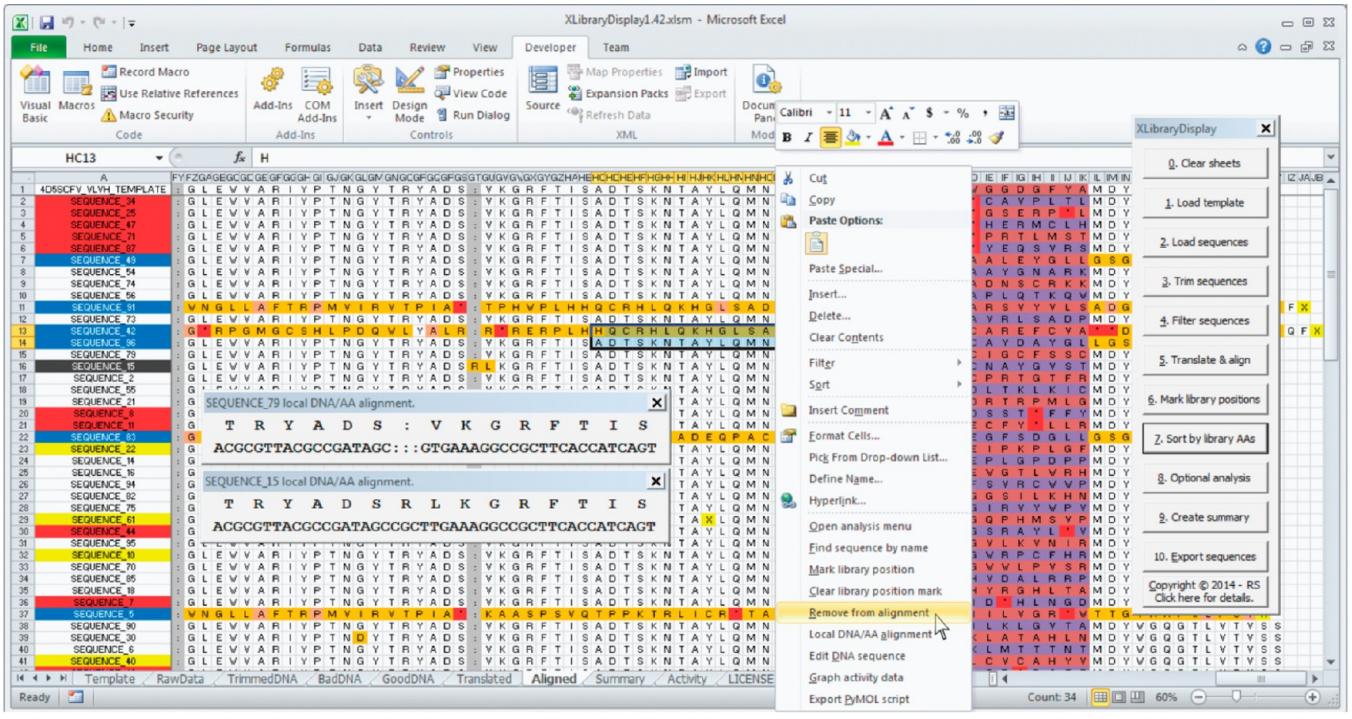


Figure 1. Overview of the XLibraryDisplay user interface. All basic analysis routines are executed by clicking through the buttons on the vertical main menu from top to bottom. The processed data from each step is organized in a series of worksheets: Template, RawData, TrimmedDNA, BadDNA, GoodDNA, Translated, Aligned, Summary, and Activity. The aligned protein sequences are shown for a sample data set in which CDR H3 of trastuzumab has been randomized with 8 NNK codons, which contain equal mixes of all nucleotides at the first two positions (N) and G or T at the third position (K). NNKs allow coding of all 20 amino acids while lowering the odds of finding stop codons in individual library members compared to NNN. The template is always visible as a reference (top row) as are the sequence names (left column) in frozen panes. Sequence names are automatically highlighted in different colors if they have stop codons (red), frameshifts (blue), deletions (gray), insertions (dark gray), or undetermined amino acids (yellow). The library positions, which were automatically detected by the program, are highlighted in magenta in the template sequence. Unique library residues within the alignment are highlighted in alternating shades of magenta and purple by default after sorting. Other amino acids in the alignment are automatically highlighted if they are mutations (orange), silent mutations (peach), stop codons (red), gaps (gray), or unknown amino acids (Xs, yellow). Right-clicking on the alignment opens an interactive menu that allows the user to perform different actions on selected sequences or columns. As an example, two local DNA/AA alignment windows are shown. The Developer tab in Excel has been enabled which allows the user to modify code using Visual Basic.

usually performed on a single DNA/protein framework. Accordingly, XLibraryDisplay has the user provide a DNA template which is used for finding the open reading frame, trimming the sequences, and also serves as a reference for performing alignments. Insertions, deletions, frameshifts, and mutations are easily identified and visualized by comparison to the same template. Once the sequences are aligned properly, the library residues can be identified, sorted, colored, counted, and correlated to experimental activity data. The program can also extract sequence and structural information from PDB files which can be aligned to the library sequences. Visualization scripts for PyMOL can then be exported for analyzing structure−activity relationships. One can also export trimmed DNA, translated proteins, or library amino acids for analysis by other programs such as WebLogo,[21] Jalview,[5] SARVisionl Biologics,[10] PFAAT,[11] or any other tool.

The analysis of raw DNA sequences is fraught with many difficulties including bad data quality, missed base calls, insertions, deletions, frameshifts, random mutations, mixed sequences, and other problems. To show how XLibraryDisplay handles these complications, an example of a recently published naïve enzyme library of the *Methanococcus jannaschii* tyrosyl tRNA synthetase (MjTyrRS) will be examined in detail here.[22] A recently published antibody library will also be described to illustrate how sequences can be correlated to other experimental and structural data.[23] Both data sets are representative of a typical protein engineering workflow that starts with a library in which specific residues are targeted for randomization—XLibraryDisplay calls these residues "library positions". To determine if library positions are randomized as intended, a number of clones are usually sequenced from the library before one starts a selection or screening campaign. All basic analysis steps for initial library QC and sequence-activity correlation are accessed through a simple GUI consisting of a main menu with buttons that the user simply clicks from top to bottom (Figure 1). Additional analysis routines are available through submenus or by clicking on aligned sequences. Different analysis options are highlighted here to show how one can apply the program to any protein library.

IMPLEMENTATION

Overall Program Architecture. XLibraryDisplay is written in VBA and integrated into a Microsoft Excel for Windows Macro-Enabled Workbook (a Windows Excel 2007, 2010, or 2013.xlsm file). The custom forms for the GUI contain minimal code so the program could be organized into various modules: 1. File loading routines (FileSubroutines). 2. Main menu routines (PrimaryAnalysis). 3. Optional analysis routines (SecondaryAnalysis). 4. Interactive menu routines (Right-ClickMenus). 5. Sequence manipulation functions (SeqFunc-

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

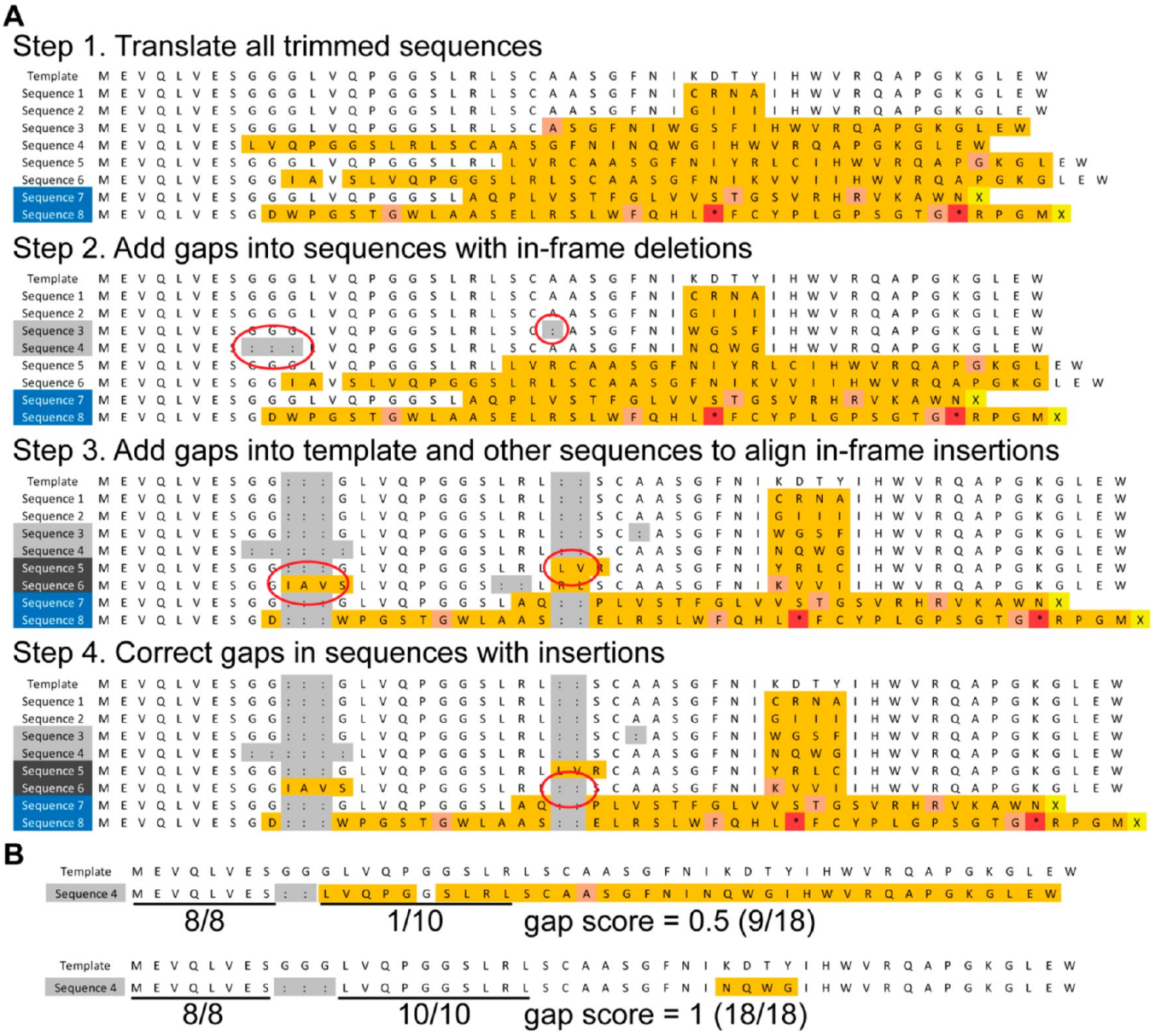


Figure 2. Simple alignment algorithm. (A) A library is shown after each step to illustrate the alignment algorithm. For simplicity, the example library only shows eight sequences from a simulated library of the trastuzumab HC in which residues KDTY of CDR H1 have been randomized with 4 NNK codons. First, all the sequences which have been previously trimmed to the template are translated and aligned from N to C terminus without adding any gaps. Second, DNA sequences 3 and 4 which are shorter than the template by a multiple of 3 are assumed to have deletions. Gaps (red circles) are inserted into these sequences to align them to the template. Third, DNA sequences 5 and 6 which are longer than the template by a multiple of 3 are assumed to have insertions. Gaps are inserted into all other sequences and the template to align these insertions (red circles). Fourth, gaps (red circle) must be corrected for sequences which also contain insertions. Lastly, DNA sequences 7 and 8 which differ in size from the DNA template by a nonmultiple of 3 are assumed to have frameshifts. No gaps are inserted into these sequences. All sequences are colored as in Figure 1. The library residues have not yet been identified, nor have the sequences been sorted so they are not colored in magenta and purple. The randomized library positions are clearly identifiable under the KDTY template residues since they are mostly mutated. (B) The simple alignment algorithm inserts gaps into sequences by systematically testing gaps and scoring up to 10 residues surrounding each gap for the best match to the template. Gaps of 1 amino acid (i.e., 1 codon or 3 nucleotides) are tested initially scanning from N- to C-terminus during the first pass. The test gap size is increased until the template and sequence are the same length. If a gap score of 1 is not found, then the gap with the highest score will be used. The gap insertion process is iterated until the sequence and template are the same length. The top example represents an intermediate gap test that did not score as well as the gap chosen in the bottom alignment. The same method is employed for inserting gaps into the template for sequences that have insertions.

tions). and 6. Miscellaneous functions (OtherFunctions). Aside from Microsoft Excel, nothing needs to be installed to run XLibraryDisplay. Once an Excel workbook is opened that contains the program, the main menu GUI will open automatically after the user enables the use of macros.

DNA Sequence Trimming. To find the site to trim on the 5′ end, each DNA sequence is searched from 5′ to 3′ for the best match to the first 20 bps of the template. To accommodate sequencing errors or mutations, sequences are trimmed if ≥90% (18/20) of the bps match by default. Similarly, to find

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

the site to trim on the 3′ end, each DNA sequence is searched from 3′ to 5′ for the best match to the last 20 bps of the template using the same criteria by default. The match and length parameters can be adjusted by the user. If only the 5′ *or* the 3′ end is trimmed during this initial pass, but not both, then the unmodified end is trimmed such that the length of the final DNA sequence is equal to the length of the template. The same procedure is automatically performed on the reverse complement of a sequence if its 5′ and 3′ ends match the template better than the original sequence. The program will indicate if sequences have been converted to their reverse complements by appending "RevComp" to the sequence name. Sequences that do not match the template by these criteria are not trimmed.

Simple Alignment Algorithm. The alignment of sequences derived from a parent sequence is a special case of multiple sequence alignment which allows a simple template-dependent alignment algorithm to be employed (Figure 2A). We call this method the simple alignment algorithm. First, all the DNA sequences which have been trimmed to the template should already be aligned at the N-terminal end when translated (sequences 1 and 2). DNA sequences that are exactly the same length of the template do not require further modification, but those that are shorter or longer than the template need to be addressed (sequences 3 through 8). These sequences are assumed to have either deletions *or* insertions. Insertions and deletions are assumed to be in-frame if the size difference relative to the template is a multiple of 3 (sequences 3 through 6). In-frame deletions are fixed by introducing gaps into the sequence to match the template length (sequences 3 and 4). In-frame insertions are fixed by introducing gaps into the template and all other sequences (sequences 5 and 6). Out-of-frame insertions and deletions are left unmodified since these are assumed to be frameshifts (sequences 7 and 8). Since the sequence length is critical for inserting gaps with this algorithm, only sequences that match the 5′ and 3′ end of the template are corrected for insertions and deletions.

To correct sequences with in-frame deletions, test gaps are inserted systematically from N- to C-terminus and scored. The test gap size starts at one amino acid and increments by one residue until the sequence and template length including gaps are equal. Test gaps are scored by counting the fraction of identical residues between the sequence and template surrounding each gap (up to 10 conserved residues on either side) (Figure 2B). A sequence will inherit the gap that yields the highest score (or first perfect 100% match) even if the sequence does not match the template length. To allow for multiple gaps at different positions, gap insertion is iterated recursively until the sequence and template length are equal. To enable gap insertion adjacent to nonconserved, randomized library positions, amino acids are ignored during gap scoring if they have a conservation score less than 85% of the average residue conservation score. The fractional conservation is calculated at each position prior to alignment using the first 1000 sequences equal to the template length. If no sequences match the template length in the data set, then all residues are considered conserved.

In-frame insertions are corrected in essentially the same manner as in-frame deletions, except gaps are inserted into the template instead of the sequence. To ensure all other sequences stay aligned to the template, gaps inserted into the template are subsequently added to all other sequences. This works correctly for all sequences except ones with insertions that have not yet been corrected. Adding gaps to other sequences that a have *second* uncorrected insertion leads to a systematic error in their placement equal to size of the *second* uncorrected insertion. This only applies if the *second* insertion lies N-terminally to the initial insertion. For example, in step 4 of Figure 2A, the circled two-residue gap of sequence 6 has been shifted *three* residues to the C-terminus to account for the error resulting from the N-terminal *three* residue insertion "IAV".

The completion time of the overall algorithm depends on the number of sequences that contain insertions and deletions. If the data set contains no insertions or deletions, then completion will be nearly instantaneous, except for the time required to (a) calculate the conservation score, (b) confirm that each sequence matches the template length, and (c) write out the alignment. Correcting each insertion or deletion will add time to completion proportional to the size of the insertion or deletion (n) and length of the sequence (m). At a maximum, if n gaps of one residue are chosen for a total of n recursions into a single sequence, gap correction will finish in approximately $n^2 \times m$ time. Single gap insertion into a single sequence without recursion will complete in approximately $n \times m$ time or less. Thus, for data sets with few insertions or deletions, the simple alignment algorithm will likely finish significantly faster than existing alignment algorithms that employ standard dynamic programming methods.

Needleman–Wunsch Alignment. XLibraryDisplay generates a multiple sequence alignment using a standard NW algorithm[24] by aligning each sibling sequence separately to the template. To ensure gaps inserted into the template are added to all sequences, each sequence is simply realigned to this template. The alignment is scored using a BLOSUM62 matrix[25] with a constant gap penalty of 10.

RESULTS

Enzyme Library Example. To illustrate how XLibraryDisplay facilitates library analysis at each step, a recently published naïve library of the MjTyrRS will be examined in detail.[22] Moreover, this will serve as a benchmark to demonstrate the accuracy of XLibraryDisplay. To briefly provide some context, the MjTyrRS was engineered to change the specificity of the enzyme from tyrosine to pAMF. Ultimately, several mutant MjTyrRS enzymes were isolated which enable expression of antibodies containing pAMF at specific positions for the production of site-specific antibody-drug conjugates. To create a screening library of MjTyrRS mutants, six residues in the active site were mutated as shown in Table 1. The library was constructed using a standard overlap extension PCR protocol[26] to introduce a roughly equal mixture of different residues at each site to give an overall theoretical library size of 1536. A total of 96 random clones were sequenced from the naïve library to assess the actual library diversity and overall quality.

Table 1. Design of the MjTyrRS Library from Zimmerman et al.[22]

position	32	65	108	109	158	159
wt	Y	L	F	Q	D	I
mutations	A	A	F	Q		A
	V	I	W	I	A	G
	L	L	Y	L	G	S
	T	V		M		V

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Thus, individual sequences will be referred to by their locations in a standard 96-well plate in this text.

To analyze these sequences with XLibraryDisplay, a template DNA file was used with 654 bps which span from Met1 through Ser218 which includes all randomized library positions. The entire protein including the C-terminal His affinity tag and stop codon is actually 312 amino acids or 939 bps. Though Sanger sequencing often affords reliable data out to more than 1000 bps, more typically reads out to ∼800 bps are the most accurate. Using a shorter template ensures the trimmed sequences represent the highest quality data which makes alignment and automated sequence analysis more reliable. During loading, the sequences were automatically detected as raw format files, text files that contain a single sequence without the name or comment line found in FASTA format. XLibraryDisplay associates the raw format file names with each loaded sequence reporting the read length and percent of assigned bases (non-Ns) on the "RawData" sheet. FASTA format files would have been automatically associated sequences with the name or comment line within each file. In this data set, A06 and G06 immediately stand out after loading for having relatively short read lengths and low percent of assigned bases, 414 bps/79.2% and 528 bps/80.9%, respectively. One can also see that all the remaining sequences have long read lengths (>1000 bps) and usually a high percent of assigned bases (mostly >90%) providing an early indication of the high data set quality.

Batch Sequence Trimming, Filtering, and Alignment. XLibraryDisplay enables the sequences to be trimmed in batch to the template and all but 3 of the 96 sequences were trimmed from the MjTyrRS library. Two of these untrimmed sequences, A06 and G06, were not trimmed because the sequencing quality is so low that there was no reliable data (confirmed by inspection of the chromatogram traces with a separate program). The other sequence that was not trimmed, E12, actually matches the vector used for cloning the library (a trastuzumab heavy chain vector with BsaI restriction sites) so it does not contain an MjTyrRS enzyme. This early analysis provides a useful measure of the library quality, indicating an estimate of the overall ligation efficiency −93 of 94 (98.9%) readable sequences match the template. Using the default settings, the three sequences that do not match the template were automatically filtered out from downstream analysis. To keep a record of sequences that are excluded from analysis, they are automatically copied to the putative "BadDNA" sheet. This does not necessarily mean these "bad sequences" have poor sequence quality (e.g., they contain unassigned bases, frameshifts, etc.). In fact E12 has high quality sequencing data, but it simply shows no match (90% sequence identity) on the 5′ or 3′ ends of the template provided so it is filtered out. At least the 5′ *or* the 3′ ends of the library sequences need to match the template in order to pass the default filters. Optional filters allow the user to exclude trimmed sequences which probably contain frameshifts (size does not equal a multiple of 3), have undetermined bases (Ns), are not the same size as the template, or if either the 5′ or the 3′ end do not match the template. All the sequences that pass the filters are transferred to the "GoodDNA" sheet.

After trimming and filtering, the user can choose from several alignment options. For this particular library, all three options (simple alignment, NW, and ClustalO[27]) align the sequences correctly, but the simple method is the fastest, enabling the entire alignment to be annotated and displayed in ∼3 s (2.6 GHz processor). The NW algorithm takes about 10 times longer (∼30 s) and ClustalO takes an intermediate time (∼10 s) on the same computer. The simple method runs the fastest because the data set contains no insertions or deletions. Fixing insertions can slow down the simple alignment algorithm significantly since all sequences need to be adjusted for each insertion. Fixing deletions does not substantially increase the algorithm completion time since each sequence is addressed in isolation. XLibraryDisplay automatically selects the simple alignment method in this case because less than half of the sequences are consistent with in-frame inserts and deletions (trimmed DNA length difference from the template *equal to* a multiple of 3). DNA sequences on the GoodDNA sheet are also adjusted by inserting gaps as necessary. Sequences on the TrimmedDNA sheet are not modified after trimming.

On the "Aligned" worksheet the wt template is placed at the top row and the sequence names in the left column in frozen panes for reference. A total of 16 sequence names are highlighted in blue to annotate probable frameshifts (5′ *and* 3′ trimmed sequences that *do not* have a length evenly divisible by 3). Another 16 sequence names are also highlighted in yellow to indicate sequences that have undetermined amino acids (highlighted in yellow Xs) which result from unrecognizable codons—usually Ns in typical DNA sequence data, but "wobble" bases, e.g. W, S, K, M, etc., are also not recognized. Mutations, silent mutations, and stop codons are also highlighted within the sequence alignment in orange, peach, and red, respectively. No insertions or deletions are in this data set, but they would have been highlighted in gray. Highlighted sequences and residues indicate what might warrant careful attention.

Interactive Sequence Analysis and Curation. Sequences with Ns or frameshifts could have been automatically removed earlier by selecting different filtering options, but it is often useful to keep these sequences in the initial analysis. In this case, it is useful to see that the majority of the annotated frameshifts (15/16) occur very close to the intended randomized positions (Table 1). For instance, six frameshifts occur within four amino acids of the randomized Y32. This suggests likely causes of these frameshifts are errors with primer annealing during the overlapping PCR protocol or errors during oligo synthesis. Manual inspection of all the chromatograms for these sequences confirms most contain real frameshifts (13/16). One incorrectly assigned frameshift, G11, actually contains a spurious G nucleotide insertion which can be corrected by clicking on the sequence. The remaining two inaccurately assigned frameshifts (D12 and D08) are actually mixed sequences (more than one DNA sequence is apparent in the chromatogram), so it is not possible to accurately analyze these sequences. Nonetheless, all the sequences marked as frameshifts except for the corrected G11 are probably best removed from further analysis by clicking on them.

One should also manually inspect the chromatograms of the 16 sequences highlighted in yellow which contain unknown amino acids. Careful inspection of their chromatograms reveals these sequences fall into three categories. In the first category, there are four sequences (B01, A02, F02, and G03) that have Ns at the beginning or end of the sequence which simply result from unassigned peaks. Certainly these sequences should be included in the analysis since there is nothing inherently wrong with them. In the second category, there are seven sequences (D03, B04, F06, B10, G10, C11, and B12) which have Ns in

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

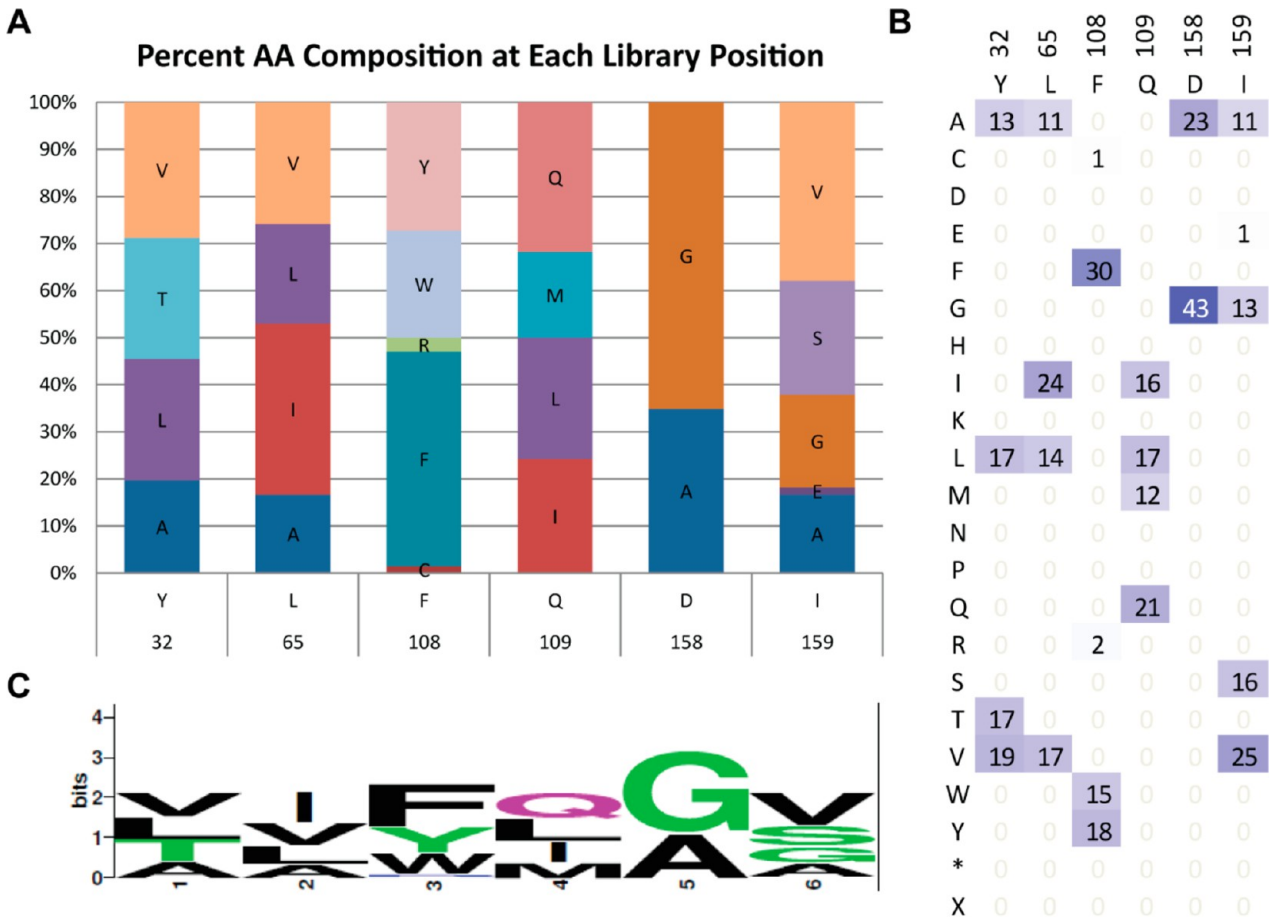


Figure 3. Automated library amino acid composition analysis. (A) A stacked-column graph generated by XLibraryDisplay shows the percent amino acid composition at each randomized position of the example MjTyrRS library. (B) A colored chart generated by the program shows the total numbers of amino acids found at each library position. (C) A WebLogo plot can be generated by loading an exported FASTA file.

the randomized library positions Q109, D158, or I159, but nowhere else in the sequence. These sequences appear to be a mixture of two or more plasmids since their chromatograms have mixtures of bases in most of the randomized library positions. Many of the mixed peaks in the chromatograms are simply assigned as single bases (not Ns). For example, position Y39 is mixed in all seven of these sequences, but single bases were assigned for the dominant peaks in all cases. This sometimes results in incorrectly translated residues not in the intended library design. For instance F108 appears to be mutated to a Cys in sequence B04, but this probably results from a simple base miscall. From one perspective, there is nothing necessarily wrong with sequences in this second category since they clearly show randomization at the intended positions, consisting of a mixture of in-frame open reading frames.

In the third category of sequences with Xs there are five clones (B07, F07, F08, A09, and F11). They are likely the result of a mixture an in-frame sequence and another sequence with a frameshift. Thus, they usually have long stretches of mutations and unknown amino acids since there are mixtures of bases throughout most of their traces and not just the randomized library positions. B07, F08, and A09 are not assigned as frameshifts by XLibraryDisplay since their 3′ ends do not match the template. F07 and F11 have 5′ and 3′ ends that match the template and were trimmed to the same size as the template so are also not assigned as frameshifts. Depending on the analysis intended, the user might exclude the second or third category. Fortunately, sequences are usually easily assigned to one of these categories by looking at the annotated alignment without having to look at sequence chromatograms.

Overall Library Composition Analysis. To obtain an accurate measure of the actual distribution of amino acids in the randomized library positions, the second and third categories of the sequences with Ns/Xs were removed. It is now clear from the alignment of these highly curated sequences that there are actually very few mutations outside of the intended randomized positions. In fact, there is only a single Y102C mutation outside of the targeted randomizations which is probably a reflection of the high fidelity Phusion polymerase used for the library construction. Since the randomized positions are obvious as columns of high percentages of mutants, the program can detect them automatically. A cutoff of 25% mutant has been arbitrarily chosen for automatic library position assignment as long as there is less than 5% Xs. However, sequences from other data sets often contain many more mutations outside of the randomized positions so automatic library detection often does not work well. In these cases, the user needs to define the randomized library positions manually. Once the library positions are assigned correctly, the distribution of amino acids at each defined library position can be counted giving a stacked column graph and color-coded chart (Figure 3A and B). One can also export the library sequences and analyze them with WebLogo[21] to visualize the sequence distribution (Figure

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

3C). Similarly, one can count the distribution of bases at each defined library position (Figure 4A and B). All analyses are consistent with the intended library design.

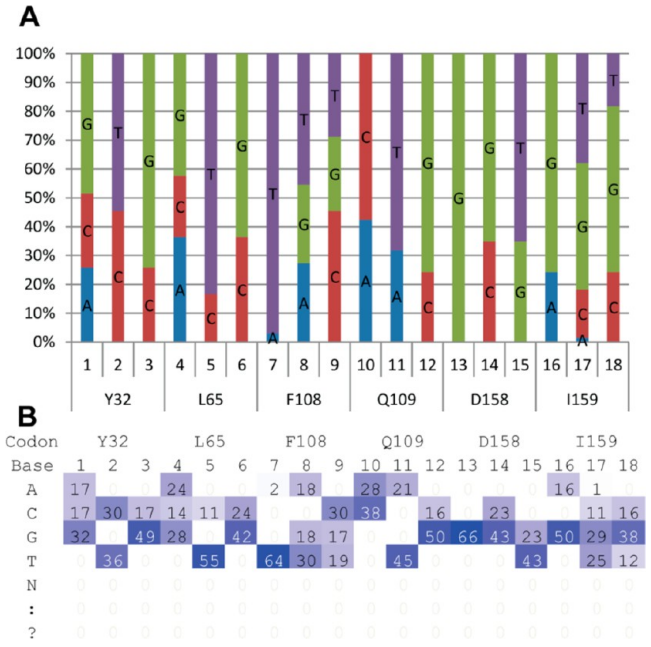


Figure 4. Automated library nucleotide composition analysis. (A) A stacked-column graph shows the percent nucleotide composition at each randomized position of the example MjTyrRS library. (B) A colored chart generated by the program shows the total numbers of each base at each library position.

Automated Library Summary Report. A summary report can be generated after the library sequences have been sorted (Figure 5). To show statistics on all sequences that contain in-frame open reading frames, only the third category of sequences with Ns were removed as well as all real frameshifts. This concise report shows that 73 of the 96 sequences (~76%) contain complete in-frame open reading frames. The actual number of good, in-frame sequences is probably slightly higher since two of the sequences were not interpretable (A06 and G06) and several sequences are likely a mixture of a frameshift and an in-frame sequence (B07, F07, F08, A09, and F11). It also reports that 70 of 73 library sequences are unique indicating the library is randomly distributed as desired. A few sequences show up in duplicate which might hint at a slight library bias, but this might also be due to bacterial replication after transformation during library QC. Alternatively, it might be expected from the relatively small library of 1536 one would see a few duplicate sequences sampled at random. From the summary, it is also clear that the wt MjTyrRS does not show up once in the library.

Phred Sequence QC Analysis. As detailed above, an analysis of sequences without looking at their chromatograms can lead to many incorrect conclusions as a result of incorrect base calls or mixed sequences. Though XLibraryDisplay cannot load or analyze sequence chromatograms directly, it can load Phred PHD formatted files as an alternative to raw or FASTA formatted sequences.[28,29] The advantage of loading PHD files is they contain a QC score for every base which can be used to visualize the accuracy of individual base calls for the entire data set (Figure 6A). Moreover, Phred scores enable XLibraryDisplay to automatically flag sequences that might be worth inspecting their individual sequence chromatograms. Specifically, after trimming the program will classify sequences as "bad data", "mixed", "not clear", "OK", or "no match, but OK".

The "bad data" classification is simply defined for any sequence that has a mean QC score less than 40 after trimming. Detecting mixed sequences is more difficult since the mean score for sequences with mixed bases can be relatively high. Mixed sequences can be distinguished from nonmixed sequences by the presence of sharp spikes of relatively low scores in the middle of the sequence (Figures 6B–D). Sequences with mean scores greater than 40 and 3 or more bad internal bases are classified as mixed. Sequences with mean scores greater than 45 and 2 or fewer bad internal bases are classified as OK. Sequences that do not fall into any of these categories are classified as "not clear". The difficulty lies in setting the bad internal score threshold and deciding which bases to define as internal. These parameters were set empirically using two independent data sets of 96 sequences

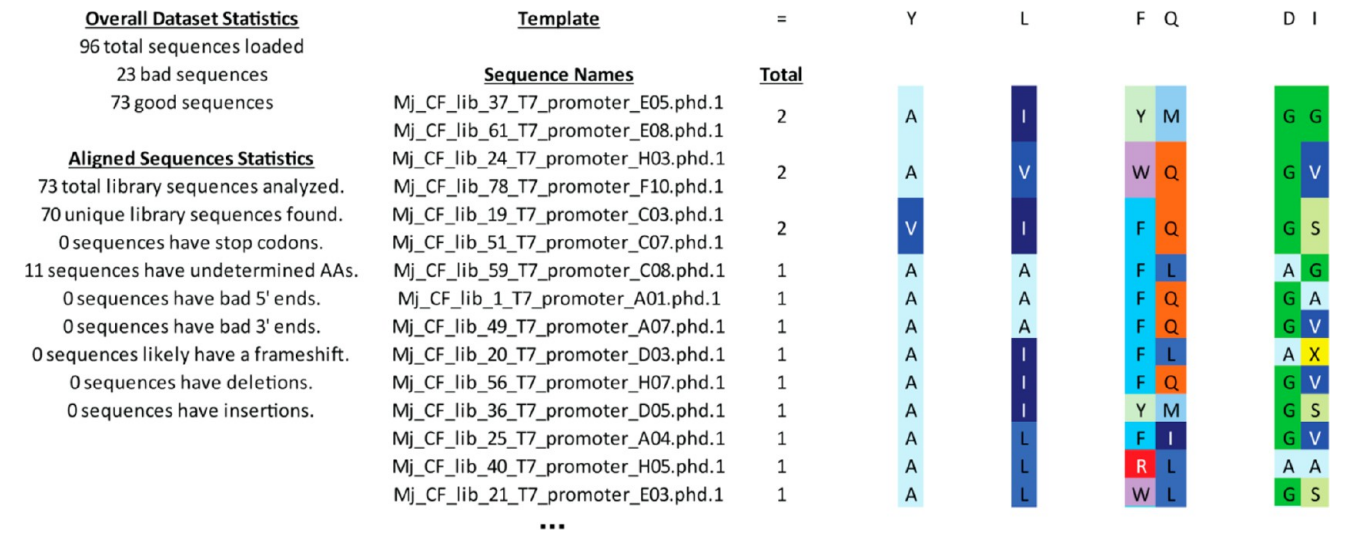


Figure 5. Library summary analysis. A portion of a standard summary report for the MjTyrRS library is shown. For brevity, only 15 sequences are shown, but the actual report shows the library sequences for all unique clones.

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX–XXX

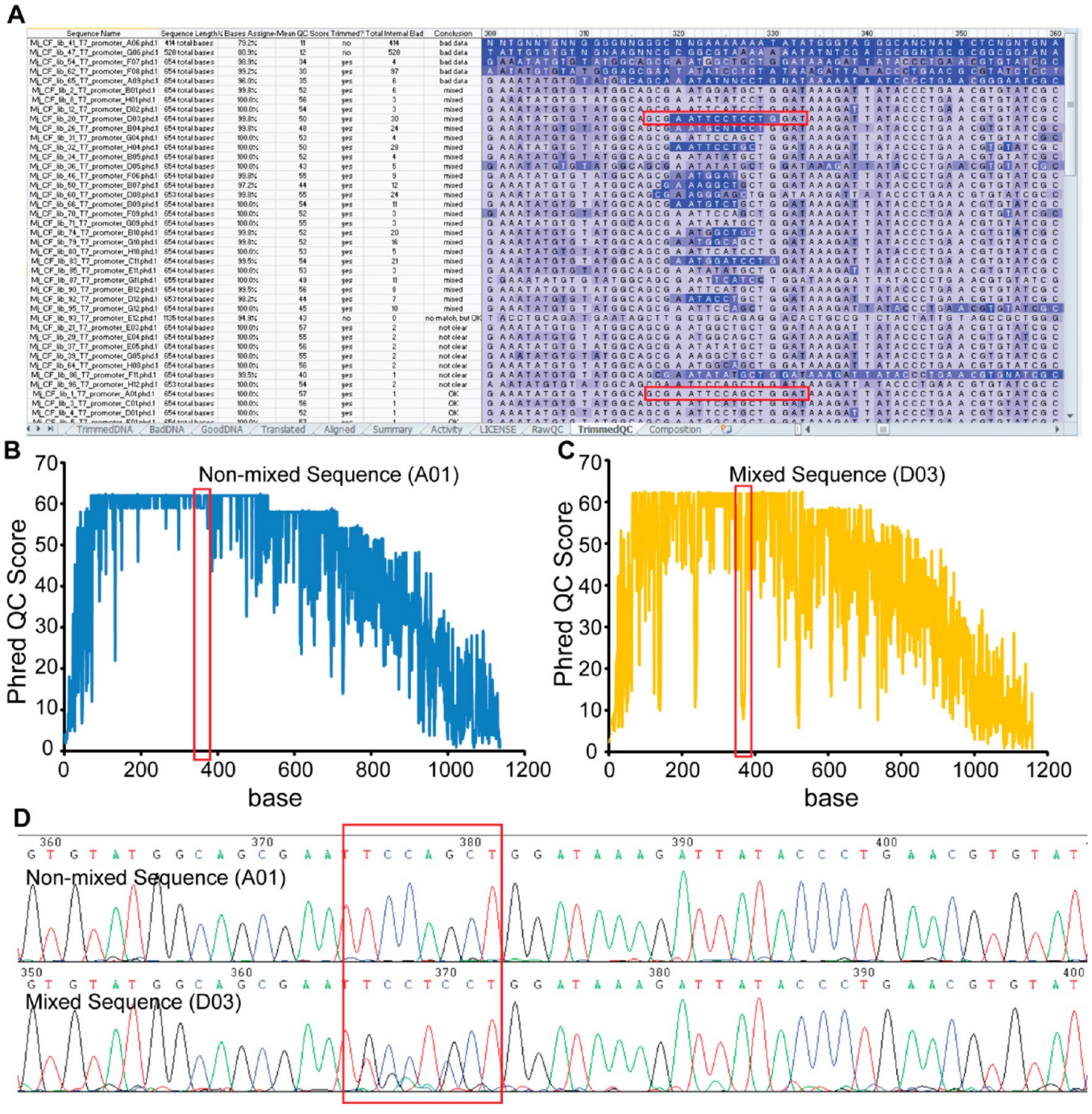


Figure 6. Phred QC analysis. (A) A typical QC report is shown for the MjTyrRS library when sequences are loaded from Phred PHD files which contain a QC score for every base. Scores for each base are used to shade separate nucleotide boxes from light blue (high score, more accurate) to dark blue (low score, less accurate). (B) A plot of the QC score at each position of a good, single sequence is shown (A01). (C) A plot of the QC score at each position of a mixture of sequences is shown (D03). (D) A comparison of the sequence chromatograms for the good sequence shown in panel B, to the mixed sequence in panel C. A red box is drawn in each panel to indicate the site of the mixed bases near position 370. The coloring in panel A enables detection of potentially mixed clones by visual inspection.

(not including the MjTyrRS library described here). Specifically, internal bases with scores less than 20 are counted as bad if they are between the first bases scoring greater than 55 reading from the 5′ and 3′ ends. The first 100 bps are ignored since they often contain sequencing artifacts that give low QC scores that are not due to mixed sequences (e.g., dye blobs). The thresholds were chosen to ensure that most mixed sequences would be detected even though that some nonmixed sequences would be incorrectly categorized as mixed. Thus, very few sequences classified as OK should actually be mixed.

Careful analysis of the chromatograms for the entire MjTyrRS data set shows excellent agreement with the Phred QC automatic sequence categorization. A total of 5 sequences are categorized as bad. These include the sequences which were automatically removed by the default filters (A06 and G06) as well as three of the sequences that appear to be a mixture of in-frame and frameshift sequences (A09, F07, and F08). The

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

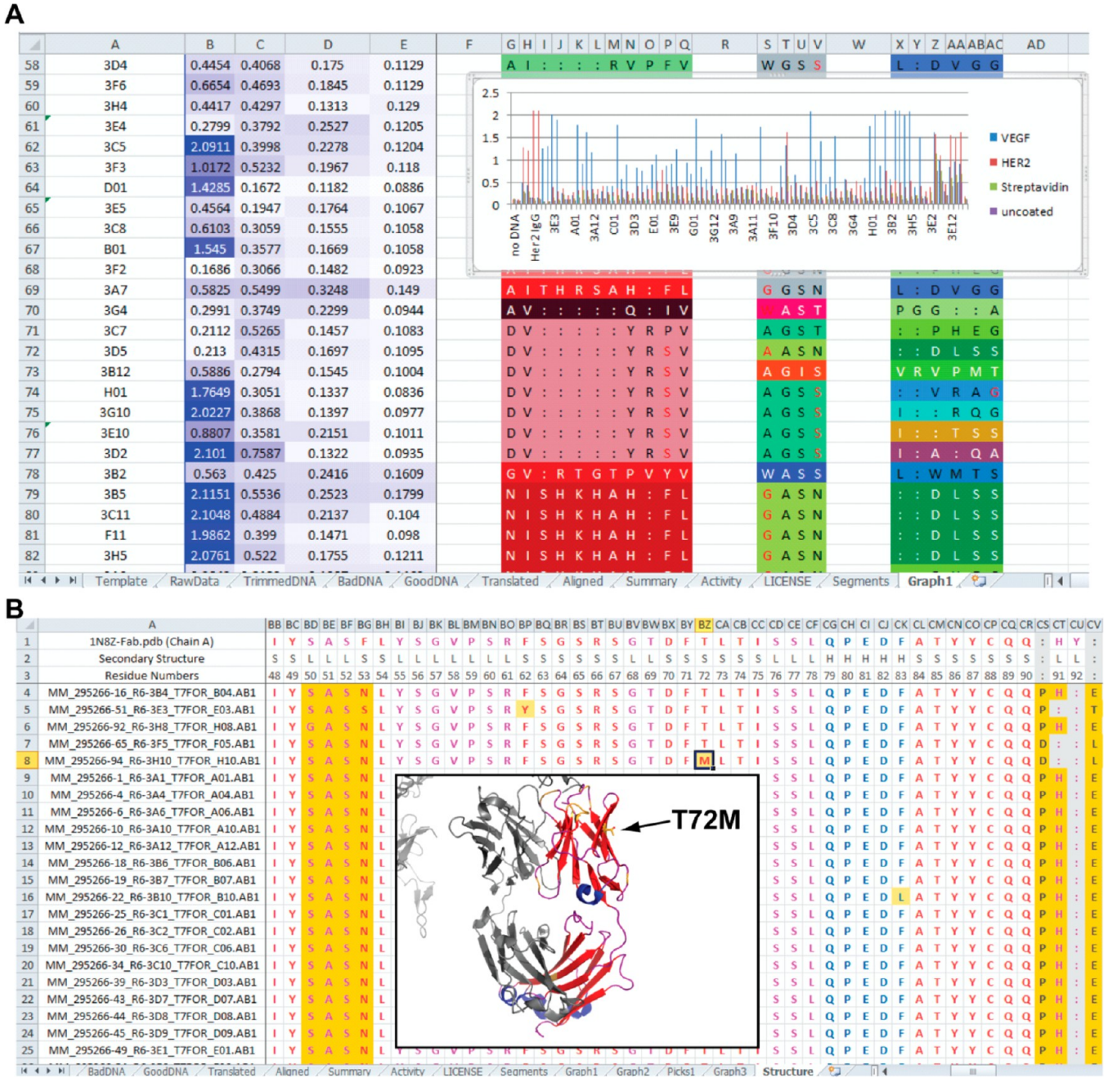


Figure 7. Activity and structure analysis. (A) An example is shown for the sequence−activity correlation for an antibody light chain library selection against VEGF. In this example, only the library sequences are shown, but the entire sequence can also be automatically correlated to experimental data. (B) An example is shown for sequence−structure correlation in which the light chain antibody sequence and structure from PDB code 1N8Z was aligned to the selected sequences shown in panel A. Residues are colored according to secondary structure (red = sheets, blue = helices, purple = loops). The inset shows an image from PyMOL created from an automatically generated script which highlights mutations like T72M.

parent trastuzumab cloning vector, E12, was also correctly classified as "no match, but OK". Of the 24 sequences categorized as mixed, 15 are actually mixed (62.5% accuracy). Of the seven sequences that were not clear two are mixed. Importantly, none of the sequences marked as OK show any obvious mixed bases. Thus, it is usually safe to not examine all the chromatograms of these OK sequences. Nonetheless, visualization of the QC scores by XLibraryDisplay often makes it easy to decide which sequence chromatograms are worth inspecting carefully. Lastly, it should be noted that most of the mixed sequences were already identified from the previous analysis of sequences with Ns (yellow Xs after translation), except for two (B09 and D05). The Phred QC analysis correctly identifies B09 and D05 as mixed even though there are no Ns in their trimmed sequences.

Activity and Structure Analysis. An important step in data analysis from a selection or screen is the correlation of experimental data to the analyzed sequences. To illustrate how XLibraryDisplay helps with this task, a different library of an antibody light-chain that was recently selected against VEGF will be described.[23] A new library is used in this case to demonstrate that other protein types can be analyzed. Antibody light-chains are relatively short (typically less than 250 amino acids), so the entire DNA sequence was used as the template. A

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

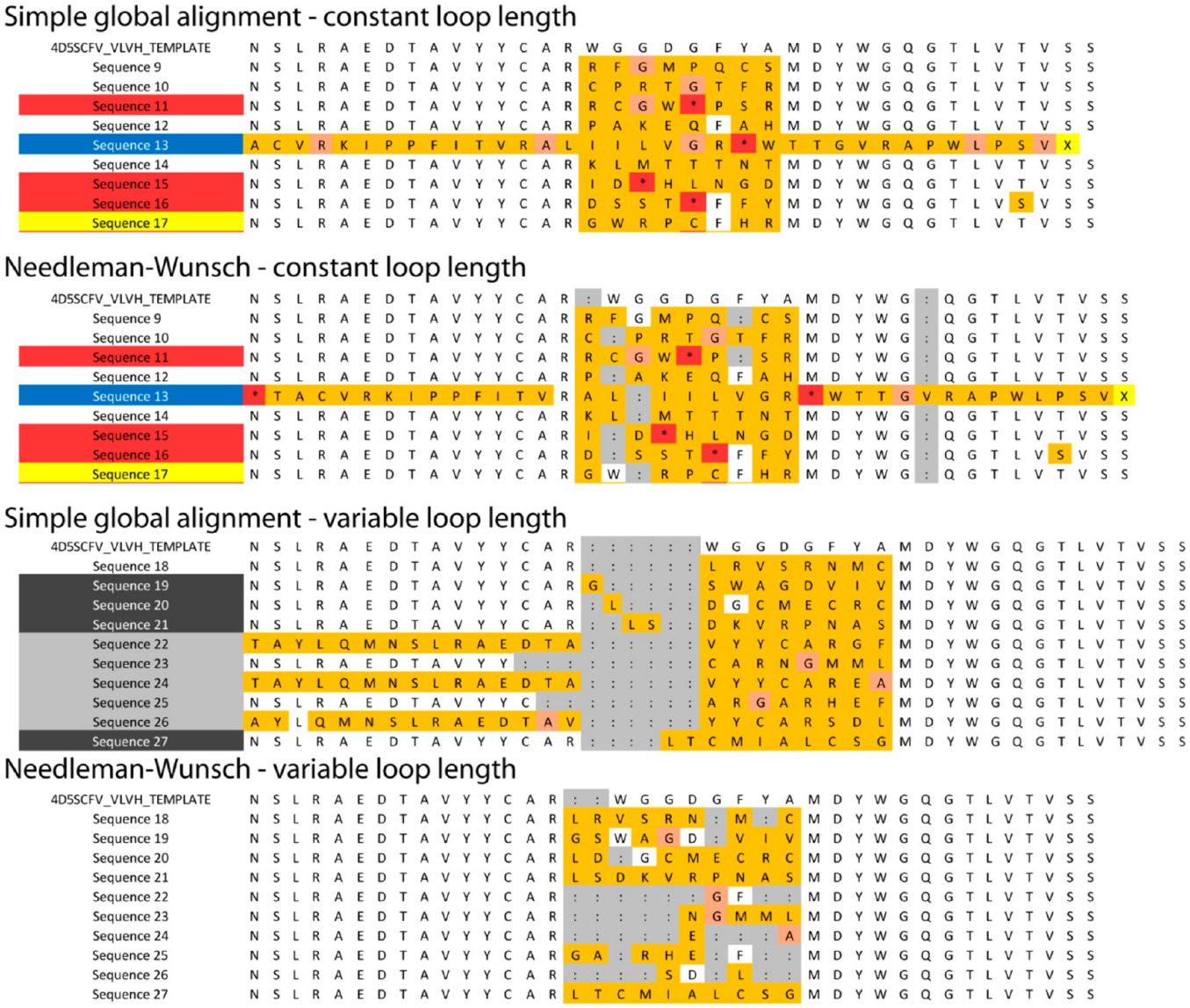


Figure 8. Comparison of alignment algorithms. For libraries with constant loop lengths, the simple alignment algorithm works better than the NW algorithm. As implemented in the program, the NW algorithm has a tendency to insert gaps into randomized positions since it uses a constant gap penalty. For libraries with variable loop lengths, the NW algorithm performs better since it correctly inserts gaps into the randomized positions.

total of 96 sequences were loaded as before for the MjTyrRS library, and 4 sequences were filtered out for having too low quality of data or not showing any match to the template (A05, D12, E11, and F07). Since this library contains loop length diversity in CDRs L1 and L3, the NW algorithm was used for alignment. After alignment, a few sequences were excluded that had obvious flaws such as frameshifts, stop codons, or other poor quality reads (B04, G07, H03, C04, B08, and G02). In fact, these clones were not included in an ELISA to assay for VEGF binding and in their place a number of controls were substituted. After data is entered onto the "Activity" sheet in a series of columns, XLibraryDisplay can then correlate it to the sequences and generate a graph (Figure 7A). The selected sequences can also be aligned to structural information which can be extracted from a standard protein data bank file (Figure 7B). This is useful for identifying where mutations acquired during the selection are in terms of the overall protein secondary structure. PyMOL scripts can also be generated to visualize in 3D the location of mutations (Figure 7B inset).

XLibraryDisplay provides other tools to help the user analyze sequence-activity relationships to help identify leads. As shown in Figure 7A, it allows different CDRs to be colored according to configurable similarity parameters and automatically highlights different residues within families. Statistics like unique CDR H3s are also reported when library segments are colored by similarity. The graph in Figure 7A usually helps the user decide where to define the cutoff for positive hits and XLibraryDisplay can then filter out negative clone based on customizable thresholds. Usually selections yield identical or similar clones multiple times and this redundancy is often reduced before proceeding with downstream characterization. Thus, XLibraryDisplay can automatically pick unique clones according to activity data for follow-up characterization.

DISCUSSION

Manual sequence analysis quickly becomes a tedious and time-consuming task for large data sets. The main design objective of XLibraryDisplay was to automate as many of these steps as

dx.doi.org/10.1021/ci500362s | J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

possible through an intuitive and flexible interface that anyone can use for any type of protein library. The generality of the program was demonstrated here through the analysis of an enzyme and an antibody library. Splitting the analysis into a series of discrete steps, rather than performing everything automatically, allows the user to exert control and quality checks at each step. This also allows users to repeat steps if they want to make changes after seeing intermediate output. For instance, one can realign sequences after a spurious frameshift is fixed. More importantly, it allows the user to make critical decisions that are not easily automated, like deciding whether a sequence containing Ns should be included in analysis or manually defining library positions.

Choosing an appropriate template is the first important step for analyzing a data set. In the example MjTyrRS library, a truncated template was used to ensure that only the highest quality sequence data was included in analysis. Actually, it usually makes sense to use the complete DNA sequence as a template first so no data is excluded from analysis. Inspection of the complete aligned sequences then allows the user to decide where to truncate the template if more accurate automated annotation is desired. For very long proteins, truncating the template is usually the better option since the data quality degenerates after ~800 bps. If the sequencing data only covers the middle portion of a protein, then it is necessary to use a template for the middle portion of the protein initially. This enables the program to trim the sequences properly since it always looks for matches to the 5′ or 3′ ends of the template to each sequence. It is important that the template is not truncated too closely to the randomized library positions since this will interfere with template matching and trimming.

The choice of alignment algorithm is another critical step for accurate analysis. For typical libraries in which most sequences are the same length as the template, the simple alignment algorithm is probably the best option. It employs a simple gap correction method to fix sequences that might acquire insertions or deletions during the directed evolution experiment. The algorithm will not properly handle individual sequences that have insertions *and* deletions, because it assumes sequences have insertions *or* deletions. This is often a reasonable assumption as spontaneous insertion and deletions rates are usually low for typical directed evolution protocols. An important exception is libraries which incorporate length variation into their design. For example, many antibody libraries use different loop lengths in CDRs to match natural repertoires—particularly CDR H3.[15] For these libraries, correcting insertions by this simple algorithm becomes cumbersome and does not work well (Figure 8). For libraries with different loop-lengths, a NW algorithm[24] is more appropriate since it effectively handles sequences that contain insertions and deletions. Thus, a simple two-pass NW algorithm has been incorporated into XLibraryDisplay for these cases. Though not necessarily the most efficient implementation, the program simply iterates twice through the alignment of each sequence to the template (i.e., a star-alignment). The NW algorithm is not recommended for many libraries because it is slow and has the tendency to insert gaps if multiple library positions are adjacent to each other (Figure 8). In principle using an affine gap penalty with a high gap opening cost could alleviate this problem,[30] but for simplicity, the NW algorithm in XLibraryDisplay uses a constant gap penalty of 10 with a BLOSUM62 scoring matrix.[25]

Instead of refining the NW implementation to enable more efficient and accurate alignment to be done internally, XLibraryDisplay simply allows the one to use a program exclusively optimized for aligning extremely large sets of sequences, ClustalO.[27] ClustalO uses an efficient hidden Markov model that should work sufficiently well for aligning most large data sets that contain insertions and deletions. Though at least one algorithm should work well in most cases, none of them can be guaranteed to yield perfect alignments for every data set. All these algorithms will occasionally inconsistently align sequences even for data sets they are best suited which can lead to analysis errors (e.g., inaccurately reporting the number of unique sequences). In general sequence alignment can be error-prone, particularly for highly divergent sequences,[31] so XLibraryDisplay allows the user to manually fix alignments by simply editing the Excel sheet. Since there are several alignment options, XLibraryDisplay tries to choose the best algorithm by default. The simple versus the NW method are chosen based on the fraction of sequences that match the template size which is indicative of length diversity in the library.

It might seem that the reliance on a single template precludes the analysis of libraries built on multiple frameworks. This is important for many antibody libraries that are built using multiple frameworks[8,32] that are not derived from the same exact reference template. It is possible to use a single representative template to analyze these libraries, however, if the template shows significant homology to the entire library as is the case for antibodies. Again, the essential requirement for using XLibraryDisplay is that at a minimum the 5′ *or* 3′ ends of the template match the library sequences. For phage display libraries, there is usually a common N-terminal leader peptide and antibodies often have a constant domain or a purification tag at the C-terminus that can also be used for template matching. The template does not necessarily need to be part of the translated protein sequence as the 5′ and 3′ untranslated regions can also be included to provide a common handle. This is an important consideration for designing a template for analyzing a library built on the extreme N- or C-terminus. The simple method will not work for the alignment of libraries built on multiple templates. Instead, the user should use the NW or ClustalO options. The library positions will likely have to be marked manually since there will likely be too many mutations from the template for automatic detection.

Certainly the most critical factor underlying a reliable analysis with XLibraryDisplay is beginning with high quality sequencing data. Even with a high quality data set, there are bound to be sequencing errors as seen in the example MjTyrRS data set. Specifically, the data set contains a spurious base call leading to an apparent frameshift that does not exist, incorrect base calls leading to incorrectly translated amino acids, numerous uncalled bases, and mixed sequences—which are usually ignored by most peak assignment programs like Phred using default parameters. The only way to ensure complete accuracy is to evaluate each sequencing chromatogram manually, but for routine library quality control this is probably unnecessary. A quick analysis with XLibraryDisplay using the automatically called bases should be sufficiently accurate to capture any major problems in library construction. The user should still be aware of the limitations of this automated analysis. This is particularly true for leads from a selection that might have sequencing errors or be a mixture of different clones. It is best that the user

carefully check sequencing chromatograms of leads which is generally good practice.[33,34]

Using modern sequencing methods it is becoming more common to have millions of sequences requiring analysis. XLibraryDisplay's integration into Microsoft Excel places limits on the number of sequences that can be analyzed. The version of Excel which XLibraryDisplay was programmed (2010) allows 1 048 576 rows that can all be used for sequence analysis and visualization in theory. Analysis has been validated for several thousand sequences, but the program still needs to be optimized for performing analysis on very large data sets (>10 000 sequences). The simple alignment algorithm might be ported into a faster language (e.g., Python or C++), however, for the analysis of high-throughput sequencing from directed evolution experiments. Excel 2010 can handle 16 384 columns and 32 767 characters per cell so there are very few proteins that are prohibitively long for analysis (theoretical limit of 10 922 amino acids). XLibraryDisplay does not currently have any features that enable stitching together multiple sequences into longer contigs, which would need to be done by another program. Some code from XLibraryDisplay also uses built-in features from Excel that also place limitations on analysis. For example, it uses Excel's sorting functionality to sort sequences by library residues which limits the total number of sortable library residues to 64 (the limit of Excel's sort fields).

The direct integration into Microsoft Excel has several advantages too. For one, it facilitates analysis and visualization with various coloring schemes, alignments, and graphs. Moreover, all the data is conveniently organized in a standard set of worksheets contained in a portable, shareable file. It also makes manual data annotation, correction, and analysis straightforward since the user can edit the automatically generated worksheets. The familiar environment lets most users quickly learn to analyze a data set with minimal instruction without fumbling through command-line program options. VBA is also a relatively easy programming language to learn, making it reasonably straightforward to add analysis routines if necessary.

Overall, the tools in XLibraryDisplay overlap somewhat with other packages that perform sequence manipulations, like the Sequence Manipulation Suite, and multiple sequence alignment analysis and annotation tools, like Jalview and PFAAT, that allow sequence-activity correlation. However, XLibraryDisplay specifically addresses problems associated with protein engineering workflows not easily handled by these programs. In particular, it helps scientists quickly triage large sets of raw DNA sequencing data for quality, redundancy, similarity, and distributions to facilitate the identification of unique, functional protein leads. PFAAT, Jalview, and SARVision|Biologics offer complementary tools once XLibraryDisplay identifies and aligns the final proteins. In particular, they offer tools not available in XLibraryDisplay like mutation cliff analysis, generation of invariant maps, linear modeling of sequence-activity relationships, calculation of phylogenetic trees, and fully integrated structural analysis.

Lastly, it should be mentioned that the GUI was loosely inspired by the crystallography software Coot, which provides a similar vertical menu for interactively refining crystal structures with many "discoverable" tools.[35] Indeed, there are additional tools in XLibraryDisplay not discussed at length in the text. These include the ability to exclude sequences by activity, look at local DNA/amino acid alignments, calculating molecular weight, extinction coefficients, and other features. Most of these additional features are available through the optional analysis menu or by right-clicking on aligned sequences. Accordingly, users are encouraged to explore different options and features beyond the basic set described here.

AUTHOR INFORMATION

Corresponding Author

*E-mail: rstafford@sutrobio.com.

Author Contributions

R.L.S. designed and wrote the program and manuscript and analyzed sequencing data. E.S.Z. designed and constructed the MjTyrRS library and analyzed sequencing data. T.J.H. and A.K.S. oversaw development of the software and library construction. All authors read and approved the manuscript.

Funding

This work was privately funded by Sutro Biopharma, Inc.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Marissa Matsumoto, Heather Stephenson, Chris Thanos, Deepti Rokkam, David Chemla-Vogel, Amandeep Gakhal, Alice Yam, Justin Diachun, and Hara Dilley for testing the software and providing suggestions for the program.

ABBREVIATIONS

NW, Needleman–Wunsch; VEGF, vascular endothelial growth factor; GUI, graphical user interface; PDB, protein data bank; MjTyrRS, *Methanococcus jannaschii* tyrosyl tRNA synthetase; pAMF, *para*-azidomethyl-L-phenylalanine; CDRs, complementarity determining regions; FW, framework; ELISA, enzyme-linked immunosorbant assay; PCR, polymerase chain reaction; VBA, Visual Basic for Applications; AA, amino acid; wt, wild type; QC, quality control; DARPin, designed ankyrin repeat protein; BLOSUM, blocks substitution matrix

REFERENCES

(1) Artimo, P.; Jonnalagedda, M.; Arnold, K.; Baratin, D.; Csardi, G.; de Castro, E.; Duvaud, S.; Flegel, V.; Fortier, A.; Gasteiger, E.; Grosdidier, A.; Hernandez, C.; Ioannidis, V.; Kuznetsov, D.; Liechti, R.; Moretti, S.; Mostaguir, K.; Redaschi, N.; Rossier, G.; Xenarios, I.; Stockinger, H. ExPASy: SIB Bioinformatics Resource Portal. *Nucleic Acids Res.* **2012**, *40*, W597–W603.

(2) Stothard, P. The Sequence Manipulation Suite: JavaScript Programs for Analyzing and Formatting Protein and DNA Sequences. *BioTechniques* **2000**, *28* (1102), 1104.

(3) Hall, T. BioEdit: A User-Friendly Biological Sequence Alignment Editor and Analysis Program for Windows 95/98/NT. *Nucleic Acids Symp. Ser.* **1999**, *41*, 95–98.

(4) Anzaldi, L. J.; Muñoz-Fernández, D.; Erill, I. BioWord: A Sequence Manipulation Suite for Microsoft Word. *BMC Bioinf.* **2012**, *13*, 124.

(5) Waterhouse, A. M.; Procter, J. B.; Martin, D. M. A.; Clamp, M.; Barton, G. J. Jalview Version 2–a Multiple Sequence Alignment Editor and Analysis Workbench. *Bioinformatics* **2009**, *25*, 1189–1191.

(6) Weiss, G. A.; Watanabe, C. K.; Zhong, A.; Goddard, A.; Sidhu, S. S. Rapid Mapping of Protein Functional Epitopes by Combinatorial Alanine Scanning. *Proc. Natl. Acad. Sci. U. S. A.* **2000**, *97*, 8950–8954.

(7) Schofield, D. J.; Pope, A. R.; Clementel, V.; Buckell, J.; Chapple, S. D.; Clarke, K. F.; Conquer, J. S.; Crofts, A. M.; Crowther, S. R.; Dyson, M. R.; Flack, G.; Griffin, G. J.; Hooks, Y.; Howat, W. J.; Kolb-Kokocinski, A.; Kunze, S.; Martin, C. D.; Maslen, G. L.; Mitchell, J. N.; O'Sullivan, M.; Perera, R. L.; Roake, W.; Shadbolt, S. P.; Vincent, K. J.; Warford, A.; Wilson, W. E.; Xie, J.; Young, J. L.; McCafferty, J.

Application of Phage Display to High Throughput Antibody Generation and Characterization. *Genome Biol.* **2007**, *8*, R254.
(8) Schwimmer, L. J.; Huang, B.; Giang, H.; Cotter, R. L.; Chemla-Vogel, D. S.; Dy, F. V.; Tam, E. M.; Zhang, F.; Toy, P.; Bohmann, D. J.; Watson, S. R.; Beaber, J. W.; Reddy, N.; Kuan, H.-F.; Bedinger, D. H.; Rondon, I. J. Discovery of Diverse and Functional Antibodies from Large Human Repertoire Antibody Libraries. *J. Immunol. Methods* **2013**, *391*, 60−71.
(9) Vielmetter, J.; Tishler, J.; Ary, M. L.; Cheung, P.; Bishop, R. Data Management Solutions for Protein Therapeutic Research and Development. *Drug Discovery Today* **2005**, *10*, 1065−1071.
(10) Hansen, M. R.; Villar, H. O.; Feyfant, E. Development of an Informatics Platform for Therapeutic Protein and Peptide Analytics. *J. Chem. Inf. Model.* **2013**, *53*, 2774−2779.
(11) Caffrey, D. R.; Dana, P. H.; Mathur, V.; Ocano, M.; Hong, E.-J.; Wang, Y. E.; Somaroo, S.; Caffrey, B. E.; Potluri, S.; Huang, E. S. PFAAT Version 2.0: A Tool for Editing, Annotating, and Analyzing Multiple Sequence Alignments. *BMC Bioinf.* **2007**, *8*, 381.
(12) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M. J. L. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics* **2009**, *25*, 1422−1423.
(13) Gentleman, R. C.; Carey, V. J.; Bates, D. M.; Bolstad, B.; Dettling, M.; Dudoit, S.; Ellis, B.; Gautier, L.; Ge, Y.; Gentry, J.; Hornik, K.; Hothorn, T.; Huber, W.; Iacus, S.; Irizarry, R.; Leisch, F.; Li, C.; Maechler, M.; Rossini, A. J.; Sawitzki, G.; Smith, C.; Smyth, G.; Tierney, L.; Yang, J. Y. H.; Zhang, J. Bioconductor: Open Software Development for Computational Biology and Bioinformatics. *Genome Biol.* **2004**, *5*, R80.
(14) Prlić, A.; Yates, A.; Bliven, S. E.; Rose, P. W.; Jacobsen, J.; Troshin, P. V.; Chapman, M.; Gao, J.; Koh, C. H.; Foisy, S.; Holland, R.; Rimsa, G.; Heuer, M. L.; Brandstätter-Müller, H.; Bourne, P. E.; Willis, S. BioJava: An Open-Source Framework for Bioinformatics in 2012. *Bioinformatics* **2012**, *28*, 2693−2695.
(15) Lee, C. V.; Liang, W.-C.; Dennis, M. S.; Eigenbrot, C.; Sidhu, S. S.; Fuh, G. High-Affinity Human Antibodies from Phage-Displayed Synthetic Fab Libraries with a Single Framework Scaffold. *J. Mol. Biol.* **2004**, *340*, 1073−1093.
(16) Lamb, B. M.; Mercer, A. C.; Barbas, C. F., 3rd Directed Evolution of the TALE N-Terminal Domain for Recognition of All 5′ Bases. *Nucleic Acids Res.* **2013**, *41*, 9779−9785.
(17) Binz, H. K.; Stumpp, M. T.; Forrer, P.; Amstutz, P.; Plückthun, A. Designing Repeat Proteins: Well-Expressed, Soluble and Stable Proteins from Combinatorial Libraries of Consensus Ankyrin Repeat Proteins. *J. Mol. Biol.* **2003**, *332*, 489−503.
(18) Koide, S.; Koide, A.; Lipovšek, D. Target-Binding Proteins Based on the 10th Human Fibronectin Type III Domain (^{10}Fn3). *Methods Enzymol.* **2012**, *503*, 135−156.
(19) Ho, M.; Pastan, I. In Vitro Antibody Affinity Maturation Targeting Germline Hotspots. *Methods Mol. Biol.* **2009**, *525*, 293−308 xiv.
(20) Labrou, N. E. Random Mutagenesis Methods for in Vitro Directed Enzyme Evolution. *Curr. Protein Pept. Sci.* **2010**, *11*, 91−100.
(21) Crooks, G. E.; Hon, G.; Chandonia, J.-M.; Brenner, S. E. WebLogo: A Sequence Logo Generator. *Genome Res.* **2004**, *14*, 1188−1190.
(22) Zimmerman, E. S.; Heibeck, T. H.; Gill, A.; Li, X.; Murray, C. J.; Madlansacay, M. R.; Tran, C.; Uter, N. T.; Yin, G.; Rivers, P. J.; Yam, A. Y.; Wang, W. D.; Steiner, A. R.; Bajad, S. U.; Penta, K.; Yang, W.; Hallam, T. J.; Thanos, C. D.; Sato, A. K. Production of Site-Specific Antibody-Drug Conjugates Using Optimized Non-Natural Amino Acids in a Cell-Free Expression System. *Bioconjugate Chem.* **2014**, *25*, 351−361.
(23) Stafford, R. L.; Matsumoto, M. L.; Yin, G.; Cai, Q.; Fung, J. J.; Stephenson, H.; Gill, A.; You, M.; Lin, S.-H.; Wang, W. D.; Masikat, M. R.; Li, X.; Penta, K.; Steiner, A. R.; Baliga, R.; Murray, C. J.; Thanos, C. D.; Hallam, T. J.; Sato, A. K. In Vitro Fab Display: A Cell-Free System for IgG Discovery. *Protein Eng. Des. Sel.* **2014**, *27*, 97−109.
(24) Needleman, S. B.; Wunsch, C. D. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *J. Mol. Biol.* **1970**, *48*, 443−453.
(25) Henikoff, S.; Henikoff, J. G. Amino Acid Substitution Matrices from Protein Blocks. *Proc. Natl. Acad. Sci. U. S. A.* **1992**, *89*, 10915−10919.
(26) Heckman, K. L.; Pease, L. R. Gene Splicing and Mutagenesis by PCR-Driven Overlap Extension. *Nat. Protoc.* **2007**, *2*, 924−932.
(27) Sievers, F.; Higgins, D. G. Clustal Omega, Accurate Alignment of Very Large Numbers of Sequences. *Methods Mol. Biol.* **2014**, *1079*, 105−116.
(28) Ewing, B.; Hillier, L.; Wendl, M. C.; Green, P. Base-Calling of Automated Sequencer Traces Using Phred. I. Accuracy Assessment. *Genome Res.* **1998**, *8*, 175−185.
(29) Ewing, B.; Green, P. Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities. *Genome Res.* **1998**, *8*, 186−194.
(30) Gotoh, O. An Improved Algorithm for Matching Biological Sequences. *J. Mol. Biol.* **1982**, *162*, 705−708.
(31) Thompson, J. D.; Plewniak, F.; Poch, O. A Comprehensive Comparison of Multiple Sequence Alignment Programs. *Nucleic Acids Res.* **1999**, *27*, 2682−2690.
(32) Tiller, T.; Schuster, I.; Deppe, D.; Siegers, K.; Strohner, R.; Herrmann, T.; Berenguer, M.; Poujol, D.; Stehle, J.; Stark, Y.; Heßling, M.; Daubert, D.; Felderer, K.; Kaden, S.; Kölln, J.; Enzelberger, M.; Urlinger, S. A Fully Synthetic Human Fab Antibody Library Based on Fixed VH/VL Framework Pairings with Favorable Biophysical Properties. *MAbs* **2013**, *5*, 445−470.
(33) Velappan, N.; Sblattero, D.; Chasteen, L.; Pavlik, P.; Bradbury, A. R. M. Plasmid Incompatibility: More Compatible than Previously Thought? *Protein Eng. Des. Sel.* **2007**, *20*, 309−313.
(34) Goldsmith, M.; Kiss, C.; Bradbury, A. R. M.; Tawfik, D. S. Avoiding and Controlling Double Transformation Artifacts. *Protein Eng. Des. Sel.* **2007**, *20*, 315−318.
(35) Emsley, P.; Lohkamp, B.; Scott, W. G.; Cowtan, K. Features and Development of Coot. *Acta Crystallogr. D Biol. Crystallogr.* **2010**, *66*, 486−501.