(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 April 2024 (11.04.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/076641 A1

(51) International Patent Classification:
G16B 40/00 (2019.01) *G16B 30/10* (2019.01)
G16B 20/00 (2019.01) *G16B 20/30* (2019.01)
G16B 30/20 (2019.01)

(21) International Application Number:
PCT/US2023/034477

(22) International Filing Date:
04 October 2023 (04.10.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/378,620 06 October 2022 (06.10.2022) US

(71) Applicant: JUST-EVOTEC BIOLOGICS, INC. [US/US]; 401 Terry Ave. N., Seattle, Washington 98109 (US).

(72) Inventors; and
(71) Applicants: **AMIMEUR, Tileli** [US/US]; 2006 Yale Ave. E., Apt. A, Seattle, Washington 98102 (US). **SHAVER, Jeremy Martin** [US/US]; 18613 41st PL NE, Lake Forest Park, Washington 98155 (US). **KETCHEM, Randal Robert** [US/US]; 810 Choctaw Ln, Shalimar, Florida 32579 (US). **TAYLOR, Alex** [US/US]; 10244 SE 16th St., Bellevue, Washington 98004 (US). **CLARK, Rutilio H.** [US/US]; 5335 Old Mill Rd NE, Bainbridge Island, Washington 98110 (US).

(74) Agent: **PERDOK, Monique M.** et al.; P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(54) **Title:** MACHINE LEARNING ARCHITECTURE TO GENERATE PROTEIN SEQUENCES

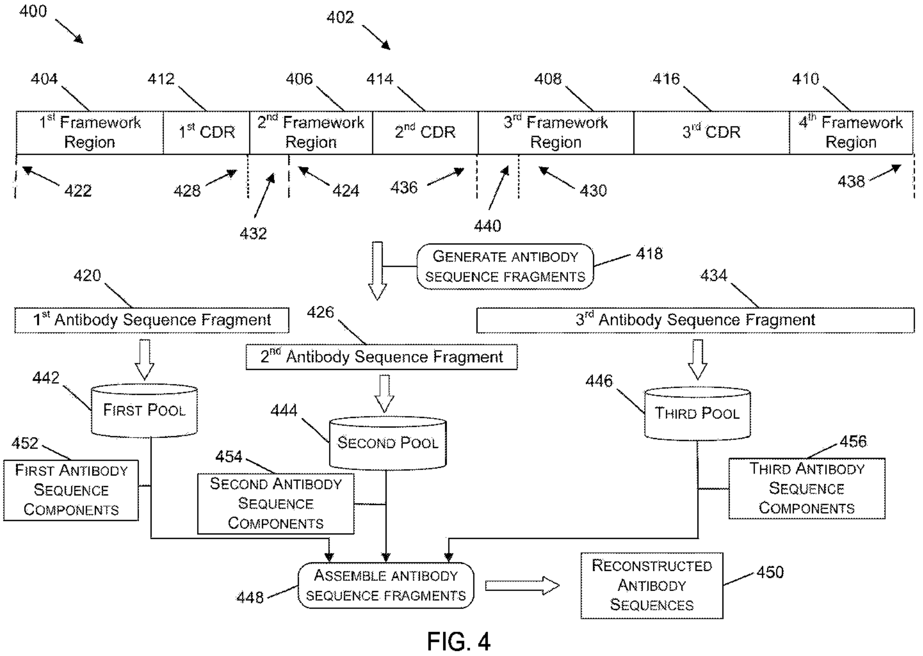

400
402
404
412
406
414
408
416
410
1st Framework Region
1st CDR
2nd Framework Region
2nd CDR
3rd Framework Region
3rd CDR
4th Framework Region
422
428
424
436
430
438
432
440
GENERATE ANTIBODY SEQUENCE FRAGMENTS
418
420
434
1st Antibody Sequence Fragment
426
3rd Antibody Sequence Fragment
2nd Antibody Sequence Fragment
442
FIRST POOL
446
THIRD POOL
444
SECOND POOL
452
FIRST ANTIBODY SEQUENCE COMPONENTS
454
SECOND ANTIBODY SEQUENCE COMPONENTS
456
THIRD ANTIBODY SEQUENCE COMPONENTS
ASSEMBLE ANTIBODY SEQUENCE FRAGMENTS
448
RECONSTRUCTED ANTIBODY SEQUENCES
450

FIG. 4

(57) **Abstract:** Amino acid sequences can be produced using a trained generative machine learning architecture. The amino acid sequences can be dividing into fragment and recombined to produce a library of new amino acid sequences. In one or more examples, the generative machine learning architecture can be trained based on amino acid sequence of at least one of antibodies or antibody segments produced by non-human mammals.

WO 2024/076641 A1

[Continued on next page]

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

MACHINE LEARNING ARCHITECTURE TO GENERATE PROTEIN SEQUENCES

BACKGROUND

[0001] Proteins are biological molecules that are comprised of one or more chains of amino acids. Proteins can have various functions within an organism. For example, some proteins can be involved in causing a reaction to take place within an organism. In other examples, proteins can transport molecules throughout the organism. In still other examples, proteins can be involved in the replication of genes. Additionally, some proteins can have therapeutic properties and be used to treat various biological conditions. The structure and function of proteins are based on the arrangement of amino acids that comprise the proteins. The arrangement of amino acids for proteins can be represented by a sequence of letters with each letter corresponding to an amino acid at a respective position. The arrangement of amino acids for proteins can also be represented by three dimensional structures that not only indicate the amino acids at various locations of the protein, but also indicate three dimensional features of the proteins, such as an α-helix or a β-sheet.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The present disclosure is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements.

[0003] Figure 1 is a diagram illustrating an example framework to generate a humanized amino acid sequence of a protein or a protein segment based on an amino acid sequence of a protein or protein segment produced by a non-human animal, in accordance with one or more implementations.

[0004] Figure 2 is a diagram illustrating an example framework to generate humanized antibody sequences from non-human antibody sequences, in accordance with one or more implementations.

[0005] Figure 3 is a diagram illustrating an example framework to perform transfer learning with respect to a generative adversarial network architecture to produce amino acid sequences of proteins having one or more specified characteristics, in accordance with one or more implementations.

[0006] Figure 4 is a diagram illustrating an example framework to divide amino acid sequences of single-domain antibodies into a number of fragments and re-assemble the fragments to produce new amino acid sequences, in accordance with one or more implementations.

[0007] Figure 5 is a diagram of a framework to analyze reconstructed protein sequences for consistency with protein sequences generated by a trained generative machine learning architecture, in accordance with one or more implementations.

[0008] Figure 6 is a flow diagram illustrating an example process to assemble amino acid sequences using fragments of additional amino acid sequences previously generated using a generative adversarial network, in accordance with one or more some implementations.

[0009] Figure 7 is a flow diagram illustrating an example process to generate reconstructed amino acid sequences using fragments of humanized antibody sequences produced using a generative machine learning architecture, in accordance with one or more implementations.

[0010] Figure 8 illustrates a diagrammatic representation of a machine in the form of a computer system within which a set of instructions may be executed for causing the machine to perform any one or more of the methodologies discussed herein, according to an example implementation.

[0011] Figure 9 illustrates an HcAb binding assessment for a number of physical antibodies having sequences generated by computational techniques corresponding to one or more example implementations.

[0012] Figure 10 illustrates luciferase activity in cells measured 24 hour post viral infection for two antibody sequences generated according to example implementations.

DETAILED DESCRIPTION

[0013] Proteins can have many beneficial uses within organisms. In particular situations, proteins can be used to treat diseases and other biological conditions that can detrimentally impact the health of humans and other mammals. In various scenarios, proteins can participate in reactions that are beneficial to subjects and that can counteract one or more biological conditions being experienced by the subjects. In some examples, proteins can also bind to target molecules within an organism that may be detrimental to the health of a subject. For these reasons, many individuals and organizations have sought to develop proteins that may have therapeutic benefits.

[0014] The development of proteins can be a time consuming and resource intensive process. Often, candidate proteins for development can be identified as potentially having various biophysical properties, structural features (e.g., negatively charged patches, hydrophobic patches), three-dimensional (3D) structures, and/or behavior within an organism. In order to determine whether the candidate proteins actually have the characteristics of interest, the proteins can be synthesized and then tested to determine whether the actual characteristics of the synthesized proteins correspond to the desired characteristics. Due to the amount of resources needed to synthesize and test proteins for specified biophysical properties, structural features, 3D structures, and/or behaviors, the number of candidate proteins synthesized for therapeutic purposes is limited. In some situations, the number of proteins synthesized for therapeutic purposes can be limited by the loss of resources that takes place when candidate proteins are synthesized and do not have the desired characteristics.

[0015] In one or more scenarios, non-human animals, such as camels, llamas, dromedaries, and alpacas, can produce heavy chain only antibodies. The antigen binding portion of these heavy chain only antibodies can be referred to as VHH antibodies, in at least some instances. In various examples, VHH antibodies can be used as therapeutics or for diagnostic applications. VHH antibodies can also be used to bind relatively small environmental chemicals, such as chemicals having molecular weights less than 1500 daltons (Da). VHH antibodies can have some advantages over polyclonal antibodies and monoclonal antibodies. For example, VHH antibodies can have increased stability when exposed to heat and some solvents and can be more soluble in water than polyclonal antibodies or monoclonal antibodies. VHH antibodies are part of a class of antibodies that can be referred to as single-domain antibodies, in some instances, or nanobodies, in other instances. Single-domain antibodies can also include heavy chain only antibodies of cartilaginous aquatic animals, such as sharks. Additionally, single-domain antibodies can be derived by separating variable regions of heavy chains or light chains of mammalian antibodies.

[0016] The techniques, methods, and systems described herein can include using generative machine learning architectures to produce amino acid sequences of proteins. In one or more examples, the amino acid sequences can correspond to protein segments. The amino acid sequences can correspond to non-human proteins and/or non-human protein segments. The amino acid sequences can be divided into fragments and then re-assembled to produce a set of new amino acid sequences. The new amino acid sequences can be analyzed to determine a measure of similarity with respect to amino acid sequences generated by the generative machine learning architecture.

[0017] In various examples, the amino acid sequences produced by the generative machine learning architectures can undergo a humanization process. For example, amino acid sequences produced by the generative machine learning architectures based on non-human proteins can be analyzed with respect to human germline amino acid sequences to determine a human germline amino acid sequence that most closely corresponds to one or more of the amino acid sequences produced by the generative machine learning architectures. A number of amino acids of the amino acid sequences produced by the generative machine learning architectures can then be modified based on the corresponding human germline amino acid sequence to generate humanized amino acid sequences. The humanized amino acid sequences can then be divided into segments and re-assembled to generate new amino acid sequences.

[0018] In at least some examples, the generative machine learning architectures can include generative adversarial networks that are trained using non-human antibody amino acid sequences. In one or more illustrative examples, the non-human antibody amino acid sequences can include amino acid sequences of single-domain antibodies. The generative adversarial networks can produce amino acid sequences that correspond to the non-human antibodies and that are then humanized based on human germline antibody amino acid sequences. The humanized antibody amino acid sequences can then be divided into fragments according to a framework that produces fragments that are overlapping in the framework regions of the humanized antibody amino acid sequences. The fragments can then be recombined to produce a large number of reconstructed humanized antibody amino acid sequences that correspond to the amino acid sequences of the non-human antibodies used to train the generative adversarial network. In various examples, the number of unpaired cysteines of the humanized antibody amino acid sequences can be minimized based on the process used to train the generative adversarial networks and/or based on the process used to humanize the non-human antibody amino acid sequences. Further, the diversity of amino acid sequences can be increased while decreasing the cost of generating a humanized antibody amino acid sequence library due to the fragmentation and re-assembly processes implemented in the techniques, methods, processes, systems, and architectures described herein.

[0019] In addition, the systems, techniques, architectures, and processes described herein can include analyzing the reconstructed antibody amino acid sequences to determine a measure of similarity to amino acid sequences produced by the generative adversarial network. In one or more examples, encoded data can be generated based on a reconstructed antibody amino acid sequence using an autoencoder. The encoded data can then be provided to the generative adversarial network to generate one or more additional antibody amino acid sequence. The reconstructed antibody amino acid sequence can then be analyzed in relation to the one or more additional antibody amino acid sequence. In situations where the measure of similarity is less than a threshold amount, the reconstructed amino acid sequence can be excluded from a library of antibody amino acid sequences.

[0020] By using amino acid sequences of non-human antibodies, additional amino acid sequences can be produced that have characteristics of the non-human antibodies, but that may also be used within a human subject. That is, the non-human antibodies may have characteristics that are not found in human antibodies, but by modifying the amino acid sequences of the non-humanized antibodies, the advantageous characteristics of the non-human antibodies can be retained while retaining their functionality within a human subject. In one or more examples, non-human antibodies can bind to one or more antigens and have increased stability in relation to human antibodies that also bind to the one or more antigens. In these situations, the amino acid sequences of the non-human antibodies can be modified to correspond to human antibodies, while retaining the increased stability of the non-human antibodies. The modified amino acid sequences can then be used to synthesize therapeutics that can be used to bind the one or more antigens in human subjects.

[0021] As used herein, structural features of proteins can refer to features of one or more amino acids or features of one or more groups of amino acids included in a protein molecule. Examples of structural features can include at least one of hydrophobic regions that include one or more amino acids, negatively charged regions that include one or more amino acids, positively charged regions that include one or more amino acids, basic regions that include one or more amino acids, acidic regions that include one or more amino acids, regions that include one or more aromatic amino acids, neutral regions that include one or more amino acids, a measure of diversity of neighboring residues, a measure of residues interacting in ionic bonds, or regions of amino acids participating in at least one of an α-helix, a β-turn, a β-sheet, or an Ω-loop. In addition, as used herein biophysical properties of proteins can refer to characteristics that can be measured with respect to a protein molecule. Examples of biophysical properties of proteins can include at least one of melting temperature, unfolding temperature, measures of aggregation, measures of stability, measures of molecular weight, measures of interactions between regions as determine by self-interaction nanoparticle spectroscopy (SINS), measures of viscosity, or measures of solubility.

[0022] Figure 1 is a diagram illustrating an example framework 100 to generate a humanized amino acid sequence of a protein or a protein fragment based on an amino acid sequence of a protein or protein fragment produced by a non-human animal, in accordance with one or more implementations. The framework 100 can include a generative machine learning architecture 102. The generative machine learning architecture 102 can implement one or more machine learning computational models to generate amino acid sequences. The amino acid sequences produced by the generative machine learning architecture 102 can correspond to proteins. In one or more examples, the sequences produced by the generative machine learning architecture 102 can include amino acid sequences of antibodies. In various implementations, the one or more machine learning computational models implemented by the generative machine learning architecture 102 can include one or more functions and one or more weights.

[0023] In one or more implementations, the generative machine learning architecture 102 can implement one or more artificial neural network technologies. For example, the generative machine learning architecture 102 can implement one or more recurrent neural networks. Additionally, the generative machine learning architecture 102 can implement one or more convolutional neural networks. Further, the machine learning architecture 102 can implement a combination of recurrent neural networks and convolutional neural networks. In one or more examples, the generative machine learning architecture 102 can include one or more generative adversarial networks (GANs). In these situations, the generative machine learning architecture 102 can include a generating component that produces amino acid sequences and a challenging component that evaluates the amino acid sequences produced by the generating component with respect to a training dataset. In further examples, the generative machine learning architecture 102 can include one or more autoencoders. In these implementations, the generative machine learning architecture 102 can include at least one of an encoder or a decoder. In one or more illustrative examples, the generative machine learning architecture 102 can include a variational autoencoder.

[0024] Protein sequence data 104 can include a number of amino acid sequences that can be used in the training of the generative machine learning architecture 102. The protein sequence data 104 can include amino acid sequences obtained from one or more data sources that store protein amino acid sequences. The protein sequence data 104 can include amino acid sequences of one or more proteins. In various examples, the protein sequence data 104 can include amino acid sequences of portions of one or more proteins. In one or more illustrative examples, the first protein sequence data 104 can include amino acid sequences of fibronectin type III (FNIII) proteins, avimers, antibodies, VHH domains, kinases, zinc fingers, combinations thereof, and the like. In one or more additional examples, the protein sequence data 104 can include amino acid sequences of portions of antibodies, such as at least a portion of one or more complementarity determining regions (CDRs) of antibodies, at least a portion of one or more light chains of antibodies, at least a portion of one or more heavy chains of antibodies, at least a portion of one or more variable regions of antibodies, at least a portion of one or more constant regions of antibodies, at least a portion of one or more hinge regions of antibodies, at least a portion of one or more antigen binding regions of antibodies, at least a portion of one or more framework regions of antibodies, one or more combinations thereof, and so forth.

[0025] In one or more implementations, the amino acid sequences included in the protein sequence data 104 used to train the generative machine learning architecture 102 can impact the amino acid sequences produced by the generative machine learning architecture 102. For example, the characteristics, biophysical properties, manufacturing characteristics (e.g., titer, yield, etc.) and so forth, of the protein sequence data 104 can impact characteristics, biophysical properties, and/or manufacturing characteristics of the amino acid sequences produced by the generative machine learning architecture 102. To illustrate, in situations where antibodies are included in the protein sequence data 104 provided to the generative machine learning architecture 102, the amino acid sequences produced by the generative machine learning architecture 102 can correspond to antibody amino acid sequences. In one or more examples, in scenarios where T-cell receptors are included in the protein sequence data 104 provided to the generative machine learning architecture 102, the amino acid sequences produced by the generative machine learning architecture 102 can correspond to T-cell receptor amino acid sequences. In one or more additional examples, in situations where kinases are included in the protein sequence data 104, the amino acid sequences produced by the generative machine learning architecture 102 can correspond to amino acid sequences of kinases. In various implementations where amino acid sequences of a variety of different types of proteins are included in the protein sequence data 104, the generative machine learning architecture 102 can generate amino acid sequences having characteristics of proteins generally and may not correspond to a particular type of protein.

[0026] During the training process, the generative machine learning architecture 102 can analyze amino acid sequences produced by the generative machine learning architecture 102 with respect to the training sequences included in the protein sequence data 104 to evaluate a loss function of the generative machine learning architecture 102. In one or more examples, output of the loss function can be used to modify the sequences generated by the generative machine learning architecture 102. For example, output related to the loss function can be used to modify one or more components of the generative machine learning architecture 102, such as an encoder, a decoder, and/or a generator of a generative adversarial network, to produce amino acid sequences that correspond more closely to amino acid sequences included in the protein sequence data 104. In one or more examples, components of the generative machine learning architecture 102 may be modified to minimize the loss function.

[0027] After the generative machine learning architecture 102 has undergone a training process, one or more trained machine learning computational models can be generated that can produce amino acid sequences of proteins. The one or more trained machine learning computational models can include one or more components of the generative machine learning architecture 102 after a training process using the protein sequence data 104. In one or more implementations, the one or more trained machine learning computational models can include a generator of a generative adversarial network that has been trained using the protein sequence data 104. Additionally, the one or more trained machine learning computational models can include at least one of an encoder or a decoder of an autoencoder of the generative machine learning architecture 102 that has been trained using the protein sequence data 104.

[0028] In one or more examples, the training process for the generative machine learning architecture 102 can be complete after the function(s) implemented by one or more components of the generative machine learning architecture 102 converge. The convergence of a function can be based on the movement of values of model parameters toward particular values as protein sequences are generated by one or more components of the generative machine learning architecture 102 and feedback is obtained in relation to at least one loss function based on differences between the amino acid sequences included in the protein sequence data 104 and the amino acid sequences generated by the generative machine learning architecture 102.

[0029] In various implementations, the training of the generative machine learning architecture 102 can be complete when the protein sequences produced by the generative machine learning architecture 102 have particular characteristics. To illustrate, the amino acid sequences generated by the generative machine learning architecture 102 can be analyzed by one or more software tools that can analyze amino acid sequences to determine at least one of biophysical properties of the amino acid sequences, structural features of the amino acid sequences, or adherence to amino acid sequences corresponding to one or more protein germlines. As used herein, germline, can correspond to amino acid sequences of proteins that are conserved when cells of the proteins replicate. An amino acid sequence can be conserved from a parent cell to a progeny cell when the amino acid sequence of the progeny cell has at least a threshold amount of identity with respect to the corresponding amino acid sequence in the parent cell. In one or more illustrative examples, a portion of an amino acid sequence of a human antibody that is part of a kappa light chain that is conserved from a parent cell to a progeny cell can be a germline portion of the antibody. In at least some examples, an amino acid sequence of a germline protein can be determined based on genomic information that corresponds to the germline cells, such as at least one of deoxyribonucleic acid (DNA) information or ribonucleic acid (RNA) information.

[0030] In one or more implementations, the analysis of amino acid sequences can be used to determine whether training of the generative machine learning architecture 102 is to cease or to continue. For example, the software tool can determine that less than a threshold amount of amino acid sequences produced by the generative machine learning architecture 102 have one or more specified characteristics. In these scenarios, the training of the generative machine learning architecture 102 may continue. Additionally, in situations where the software tool determines that a threshold amount of amino acid sequences produced by the generative machine learning architecture 102 do correspond to one or more specified characteristics, the training of the generative machine learning architecture 102 can stop.

[0031] The protein sequences included in the protein sequence data 104 can be subject to data preprocessing 106 before being provided to the generative machine learning architecture 302. In one or more implementations, the protein sequence data 104 can be arranged according to a classification system, also referred to as a classification schema, before being provided to the generative machine learning architecture 102. The data preprocessing 106 can include pairing amino acids included in the proteins of the protein sequence data 104 with numerical values that can represent structure-based positions within the proteins. The numerical values can include a sequence of numbers having a starting point and an ending point. In an illustrative example, a T can be paired with the number 43 indicating that a Threonine molecule is located at a structure-based position 43 of a specified protein domain type. In one or more illustrative examples, structure-based numbering can be applied to any general protein type, such as fibronectin type III (FNIII) proteins, avimers, antibodies, VHH domains, kinases, zinc fingers, and the like.

[0032] In one or more implementations, the classification system implemented by the data preprocessing 106 can designate a particular number of positions for certain regions of proteins.

For example, the classification system can designate that portions of proteins have particular functions and/or characteristics can have a specified number of positions. In various situations, not all of the positions included in the classification system may be associated with an amino acid because the number of amino acids in a specified region of a protein may vary between proteins. To illustrate, the number of amino acids in a region of a protein can vary for different types of proteins. In one or more examples, positions of the classification system that are not associated with a particular amino acid can indicate various structural features of a protein, such as a turn or a loop. In an illustrative example, a classification system for antibodies can indicate that heavy chain regions, light chain regions, and hinge regions have a specified number of positions assigned to them and the amino acids of the antibodies can be assigned to the positions according to the classification system.

[0033] The output produced by the data preprocessing 106 can include structured sequences 108. The structured sequences 108 can include a matrix indicating amino acids associated with various positions of a protein. In one or more examples, the structured sequences 108 can include a matrix having columns corresponding to different amino acids and rows that correspond to structure-based positions of proteins. For each element in the matrix, a 0 can be used to indicate the absence of an amino acid at the corresponding position and a 1 can be used to indicate the presence of an amino acid at the corresponding position. In situations where a position represents a gap in an amino acid sequence, the row associated with the position can comprise zeroes for each column. The sequence(s) analyzed and/or generated by the generative machine learning architecture 102 can also be represented using a vector according to a same or similar number scheme as used for the structured sequences 108. In at least some illustrative examples, the structured sequences 108 and sequence(s) analyzed by and/or generated by the generative machine learning architecture 102 can be encoded using a method that may be referred to as a one-hot encoding method.

[0034] The training process for the generative machine learning architecture 102 can produce a trained generative machine learning component 110. The trained generative machine learning component 110 can include one or more machine learning computational models. In various examples, the one or more machine learning computational models of the trained generative machine learning component 110 can generate sequences of amino acids of proteins that correspond to the amino acid sequences included in the protein sequence data 104. In one or more examples, the trained generative machine learning component 110 can include a generating component of a generative adversarial network. In one or more additional examples, the trained generative machine learning component 110 can include a trained generator of a generative adversarial network. In one or more further examples, the trained generative machine learning component 110 can include at least one of an encoder or a decoder of an autoencoder of the generative machine learning architecture 102.

[0035] The framework 100 can also include a protein sequence assembly system 112. The trained generative machine learning component 110 can be a part of the protein sequence assembly system 112. For example, the protein sequence assembly system 112 can include a protein sequence generating system 114 that includes the trained generative machine learning component 110. The protein sequence generating system 114 can produce protein sequences 116. In one or more examples, the protein sequence generating system 114 can include a generator of a trained generative adversarial network that produces the protein sequences 116. The protein sequences 116 can include amino acid sequences of proteins or protein segments.

[0036] The protein sequence assembly system 112 can include a protein sequence fragmentation system 118 that obtains the protein sequences 116. The protein sequence fragmentation system 118 can divide individual amino acid sequence of the proteins sequences 116 into protein sequence fragments 120. The protein sequence fragments 120 can include fragments of amino acid sequences included in the protein sequences 116. In one or more examples, the protein sequence fragmentation system 118 can divide individual protein sequences of the protein sequences 116 into a number of fragments according to one or more fragmentation schemes 122. The one or more fragmentation schemes 122 can include a number of rules for dividing individual amino acid sequences of the protein sequences 116 into multiple fragments. For example, the one or more fragmentation schemes 122 can indicate one or more positions at which the protein sequence fragmentation system 118 can divide individual amino acid sequences included in the protein sequences 116. In one or more additional examples, the one or more fragmentation schemes 122 can indicate one or more ranges of positions at which the protein sequence fragmentation system 118 can divide individual amino acid sequences included in the protein sequences 116. In one or more further examples, the one or more fragmentation schemes 122 can indicate one or more regions of proteins in which the protein sequence fragmentation system 118 can divide individual amino acid sequences included in the protein sequences 116.

[0037] In one or more illustrative examples where the protein sequences 116 correspond to amino acid sequences of antibodies, the one or more fragmentation schemes 122 can indicate that the protein sequences 116 are to be divided into fragments in the framework regions of the antibodies. In one or more additional illustrative examples where the protein sequences 116 correspond to amino acid sequences of antibodies, the one or more fragmentation schemes 122 can indicate that positions of the amino acid sequences included in complementarity determining regions (CDRs) of the antibodies are not to be used as points where the amino acid sequences can be cleaved by the protein sequence fragmentation system 118 to generate one or more protein sequence fragments 120. In one or more further examples, the one or more fragmentation schemes 122 can indicate that amino acid sequences included in the protein sequence fragments 120 that are generated from a same protein sequence 116 are to have an amount of overlap. The amount of overlap can include at least one amino acid, at least two amino acids, at least three amino acids, at least four amino acids, at least five amino acids, at least six amino acids, at least seven amino acids, at least eight amino acids, at least nine amino acids, at least ten amino acids, at least twelve amino acids, at least fifteen amino acids, at least eighteen amino acids, at least twenty amino acids, or at least twenty-five amino acids.

[0038] The protein sequence assembly system 112 can also include a protein sequence assembly system 124. The protein sequence assembly system 124 can combine a number of the protein sequence fragments 120 to generate reconstructed protein sequences 126. At least a portion of the reconstructed protein sequences 126 can include amino acid sequences that are different from the amino acid sequences included in the protein sequences 116. In various examples, the protein sequence assembly system 124 can implement one or more rules and/or one or more schema to generate the reconstructed protein sequences 126. For example, the protein sequence assembly system 124 can exclude amino acid sequences produced by the protein sequence assembly system 124 that correspond to an amino acid sequence included in the protein sequences 116 from the reconstructed protein sequences 126. In one or more examples, the protein sequence assembly system 124 can determine an amount of similarity between one or more reconstructed protein sequences 126 and one or more protein sequences 116 by determining an amount of homology between the one or more reconstructed protein sequences 126 and the one or more protein sequences 116. In one or more additional examples, the protein sequence assembly system 124 may be prohibited from combining protein sequence fragments 120 that originated from the same protein sequence 116. An amount of homology between amino acid sequences can be determined using an amino acid sequence alignment tool, such as SIM sequence alignment tool, a basic local alignment search tool (BLAST), or a Clustal alignment tool.

[0039] In at least some examples, the protein sequence assembly system 124 can determine an individual protein sequence 116 that corresponds to the protein sequence fragments 120 based on metadata generated by the protein sequence fragmentation system 118 during the production of the protein sequence fragments 120. In one or more examples, the protein sequence fragmentation system 118 can generate an identifier of individual protein sequences 116 and cause the identifier to be associated with the protein sequence fragments 120 that correspond to the individual protein sequences 116. For example, the protein sequence fragments 120 can be stored in memory of one or more computing devices in conjunction with the identifiers of the protein sequences 116 from which the protein sequence fragments 120 were generated. The memory can include cache memory, disk memory, flash memory, solid state memory, or other data storage devices. In various examples, the identifiers of the individual protein sequences 116 generated by the protein sequence fragmentation system 118 can include one or more alphanumeric symbols that uniquely identify the individual protein sequences 116 used by the protein sequence fragmentation system 118 to generate the protein sequence fragments 120. In one or more illustrative examples, the identifiers of the individual protein sequences 116 can include at least one amino acid sequence that is appended to the protein sequence fragments 120 that correspond to the initial individual protein sequences 116 used by the protein sequence fragmentation system 118 to generate the protein sequence fragments 120.

[0040] The framework 100 can include a protein sequence analysis system 128 that analyzes the reconstructed protein sequences 126. In one or more examples, the protein sequence analysis system 128 can determine whether at least a portion of the reconstructed protein sequences 126 correspond to the protein sequences 116. In various examples, the sequence analysis system 128 analyzes the reconstructed protein sequences 126 to determine at least a portion of the reconstructed protein sequences 126 to include in a library of protein sequences having one or more characteristics. The one or more characteristics can correspond to one or more structural features of proteins. In one or more additional examples, the one or more characteristics can correspond to one or more molecular features of proteins. The one or more characteristics can also correspond to features of one or more types of proteins. To illustrate, the protein sequences 116 can include amino acid sequences of antibodies and the sequence analysis system 128 can analyze the reconstructed protein sequences 126 to determine whether the reconstructed protein sequences 126 also correspond to amino acid sequences of antibodies. For example, the sequence analysis system 128 can determine whether the reconstructed protein sequences 126 have characteristics of antibodies, such as characteristics of at least one of one or more heavy chains, characteristics of one or more light chains, characteristics of one or more hinge regions, characteristics of one or more antigen binding regions, characteristics of one or more complementarity determining regions, or characteristics of one or more framework regions. The sequence analysis system 128 can generate an analysis output 130 that indicates whether a given reconstructed protein sequence 126 has one or more characteristics. In various examples, the sequence analysis system 128 can generate an analysis output 130 that indicates a measure of similarity between at least a portion of an individual reconstructed protein sequence 126 and at least a portion of the protein sequences 116.

[0041] Based on the analysis output 130 generated by the sequence analysis system 128, a library of amino acid sequences can be determined from among the reconstructed protein sequences 126. For example, at least a portion of the reconstructed protein sequences 126 can be selected to be included in a library of protein sequences based on the analysis output 130 for the selected reconstructed protein sequences 126 corresponding to one or more criteria. To illustrate, a library of protein sequences can be determined by identifying reconstructed protein sequences 126 having an analysis output 130 that satisfies a threshold measure of similarity in relation to the protein sequences 116. In one or more examples, a library of protein sequences can include thousands, up to tens of thousands, up to hundreds of thousands or more amino acid sequences. In one or more illustrative examples, a library of protein sequences generated by the protein sequence assembly system 112 can include at least 1000 amino acid sequences, at least 5000 amino acid sequences, at least 10,000 amino acid sequences, at least 25,000 amino acid sequences, at least 50,000 amino acid sequences, at least 100,000 amino acid sequences, at least 250,000 amino acid sequences, at least 500,000 amino acid sequences, at least 750,000 amino acid sequences, or at least 1,000,000 amino acid sequences. In various examples, proteins that correspond to the amino acid sequences included in a library of protein sequences can be synthesized. In one or more illustrative examples, proteins having amino acid sequences included in a library of protein sequences that includes at least a portion of the reconstructed protein sequences 126 can be synthesized according to the methods and techniques described in Nilsson BL, Soellner MB, Raines RT. Chemical synthesis of proteins. Annu Rev Biophys Biomol Struct. 2005;34:91-118. doi: 10.1146/annurev.biophys.34.040204.144700. PMID: 15869385; PMCID: PMC2845543. In situations where the library of protein sequences includes antibodies or antibody fragments, antibodies can be synthesized according to at least a portion of the reconstructed protein sequences 126 included in the library according to the methods and techniques described in Dangi AK, Sinha R, Dwivedi S, Gupta SK and Shukla P

(2018) Cell Line Techniques and Gene Editing Tools for Antibody Production: A Review. *Front. Pharmacol.* 9:630. doi: 10.3389/fphar.2018.00630.

[0042] In one or more examples, antibodies or antibody segments synthesized according to at least a portion of the reconstructed protein sequences 126 can bind to a given target antigen and used directly as a therapeutic in the treatment of a biological condition present in one or more patients. In scenarios where a library that includes at least a portion of the reconstructed protein sequences 126 includes antibody segments, one or more therapeutics can be produced by combining the antibody segments with one or more additional antibody segments. For example, a library of VHH antibodies can be combined with fragment crystallizable regions (Fc) to produce one or more therapeutics used in the treatment of a biological condition. In one or more additional examples, a library of at least a portion of the reconstructed protein sequences 126 can be used to synthesize T-cells that can be used in chimeric antigen receptor (CAR) T-cell therapy to treat one or more biological conditions.

[0043] Although not explicitly shown in the illustrative example of Figure 1, the framework 100 can include one or more systems and/or one or more components that can perform computational operations to humanize amino acid sequences of proteins that are produced by non-human animals. For example, the amino acid sequences included in the protein sequence data 104 can include amino acid sequences that were humanized based on amino acid sequences of proteins produced by non-human animals. In one or more examples, amino acid sequences of proteins produced by non-human animals can be generated by determining a human germline amino acid sequence that corresponds to an amino acid sequence of a protein produced by a non-human animal. At least a portion of the positions of the amino acid sequences of the protein produced by a non-human animal can be modified based on amino acids located in one or more positions of the human germline amino acid sequence. The generative machine learning architecture 102 can then be trained using the humanized amino acid sequences included in the protein sequence data 104.

[0044] In one or more additional examples, the reconstructed protein sequences 126 can undergo a humanization process in situations where the protein sequence data 104 includes amino acid sequences of proteins produced by non-human animals. In these scenarios, a human germline amino acid sequence that corresponds to one or more of the reconstructed protein sequences 126 can be determined. Amino acids located at a portion of the positions of the reconstructed protein sequences 126 can be modified based on the amino acids located at one or more corresponding positions of the human germline amino acid sequence to generate a humanized protein amino acid sequence. In various examples, the humanized protein amino acid sequences can then be analyzed by the sequence analysis system 128 in relation to one or more characteristics.

[0045] Figure 2 is a diagram illustrating an example framework 200 to generate humanized antibody sequences from non-human antibody sequences, in accordance with one or more implementations. The framework 200 can include an antibody humanization system 202 that modifies amino acids located at one or more positions of non-human antibody sequences 204 to generate humanized antibody sequences 206. The non-human antibody sequences 204 can correspond to amino acid sequences of at least one of antibodies or antibody segments produced by non-human animals. In one or more examples, the non-human antibody sequences 204 can correspond to amino acid sequences of at least one of antibodies or antibody segments produced by one or more camelids. The one or more camelids can include at least one of camels, llamas, alpacas, dromedaries, guanacos, or vicuñas. In one or more additional examples, the non-human antibody sequences 204 can correspond to amino acid sequences of at least one of antibodies or antibody segments produced by cartilaginous fish, such as sharks. In various examples, an antibody segment can correspond to at least a portion of an antibody. For example, an antibody segment can correspond to at least a portion of an antibody light chain or at least a portion of an antibody heavy chain. In one or more additional examples, an antibody segment can correspond to an antigen binding region of an antibody light chain or an antigen binding region of an antibody heavy chain. In one or more further examples, an antibody segment can correspond to at least one of one or more complementarity determining regions (CDRs) of an antibody or one or more framework regions of an antibody. In various examples, an antibody segment can correspond to a domain of an antibody.

[0046] In one or more illustrative examples, the non-human antibody sequences 204 can include amino acid sequences of single-domain antibodies produced by a non-human animal. The single-domain antibodies can comprise an antibody segment having a single variable region antibody domain that can bind to one or more antigens. In various examples, the non-human antibody sequences 204 can include amino acid sequences of VHH segments produced by one or more camelids. In one or more further examples, the non-human antibody sequences 204 can include amino acid sequences of VNAR segments produced by one or more cartilaginous fish. The non-human antibody sequences 204 can correspond to single-domain antibodies having a molecular weight no greater than about 20 kilodaltons (kDa), no greater than about 18 kDa, no greater than about 15 kDa, no greater than about 12 kDa, no greater than about 10 kDa, or no greater than about 8 kDa. In at least some examples, the non-human antibody sequences 204 can correspond to single-domain antibodies having a molecular weight from about 8 kDa to about 20 kDa, from about 10 kDa to about 18 kDa, or from about 12 kDa to about 15 kDa.

[0047] In one or more examples, the antibody humanization system 202 can analyze one or more non-human antibody sequences 204 in relation to one or more template sequences 208. The one or more template sequences 208 can include amino acid sequences of antibodies and/or amino acid sequences of antibody segments that can provide substitute amino acids that can replace amino acids at one or more positions of the one or more non-human antibody sequences 204 to generate the humanized antibody sequences 206. In one or more examples, the template sequences 208 can include human antibody germline sequences 210. The human antibody germline sequences 210 can include amino acid sequences of antibodies and/or amino acid sequences of antibody segments produced according to genomic material included in human germline cells.

[0048] The antibody humanization system 202 can analyze individual non-human antibody sequences 204 with respect to a number of human antibody germline sequences 210 to determine a measure of similarity between the individual non-human antibody sequences 204 and the number of human antibody germline sequences 210. In various examples, the antibody humanization system 202 can determine quantitative measures that indicate an amount of similarity between individual non-human antibody sequences 204 and one or more human antibody germline sequences 210. In one or more examples, the antibody humanization system 202 can determine the quantitative measures by determining an amount of homology between individual non-human antibody sequences 204 and one or more human antibody germline sequences 210. In at least some examples, for each non-human antibody sequence 204, the antibody humanization system 202 can determine a set of quantitative measures, with individual quantitative measures indicating an amount of similarity between the non-human antibody sequence 204 and an individual human antibody germline sequence of the number of the human antibody germline sequences 210. The antibody humanization system 202 can determine the human antibody germline sequence 210 having a highest quantitative measure in the set of quantitative measures. The antibody humanization system 202 can then modify amino acids at one or more positions of the non-human antibody sequence 204 based on amino acids located at one or more positions of the human antibody germline sequence 210 having the highest quantitative measure in the set of quantitative measures. In one or more illustrative examples, the human antibody germline sequences 210 can correspond to at least one of the HV323 germline gene or the HV330 germline gene.

[0049] In one or more illustrative examples, the antibody humanization system 202 can determine a human antibody germline sequence 210 that corresponds to a non-human antibody sequence 204 in a piecewise fashion. For example, pieces of a human antibody germline sequence 210 can be compared with pieces of a non-human antibody sequence 204 to determine an amount of homology between the respective pieces. The amount of homology between the respective pieces of the human antibody germline sequence 210 and the non-human antibody sequence 204 can be combined to determine an overall quantitative measure indicating an amount of homology between the human antibody germline sequence 210 and the non-human antibody sequence 204. The quantitative measures of multiple human antibody germline sequences 210 with respect to the non-human antibody sequences 204 can be ranked with a human antibody germline sequence 210 having the highest rank being selected by the antibody humanization system 202 and used to generate one or more humanized antibody sequences 206 based on a respective non-human antibody sequence 204.

[0050] The antibody humanization system 202 can generate the humanized antibody sequences by modifying one or more positions of individual non-human antibody sequences 204 according to one or more template sequences 208 and according to sequence modification data 212. The sequence modification data 212 can include at least one of one or more schemes, one or more rules, or one or more frameworks to use in the modification of amino acids of the individual non-human antibody sequences 204 to generate the humanized antibody sequences 206. For example, the sequence modification data 212 can indicate one or more positions of the non-human antibody sequences 204 that are to be conserved and not modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. To illustrate, the sequence modification data 212 can indicate that one or more regions of one or more complementarity determining regions (CDRs) of the non-human antibody sequences 204 are to be conserved and not modified to generate the humanized antibody sequences 206. The sequence modification data 212 can also indicate that one or more positions that correspond to a light chain and heavy chain interface of the non-human antibody sequences 204 are to be conserved and not modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. Additionally, the sequence modification data 212 can indicate a number of positions in framework regions of the non-human antibody sequences 204 that can be modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. Further, the sequence modification data 212 can indicate an upper threshold and/or a lower threshold corresponding to a number of amino acids that can be modified in one or more regions of the non-human antibody sequences 204 to generate the humanized antibody sequences 206.

[0051] In one or more examples, the sequence modification data 212 can indicate one or more rules or one or more schema to increase the likelihood of disulfide bond formation between cysteine molecules included in amino acid sequences of the humanized antibody sequences 206. In at least some examples, the sequence modification data 212 can indicate that the antibody humanization system 202 is to minimize the number of unpaired cysteines that may be present in the humanized antibody sequences 206. In various examples, the sequence modification data 212 can indicate at least one of a number of positions in which intra-chain cysteines are to be paired or a number of positions in which inter-chain cysteines are to be paired. The minimization of unpaired cysteines in the humanized antibody sequences 206 can increase the likelihood that an antibody that is synthesized according to a humanized antibody sequence 206 can maintain one or more structural features. In at least some cases, maintaining the one or more structural features of the antibodies synthesized from the humanized antibody sequences 206 can increase a likelihood of the antibodies binding to one or more target antigens. Additionally, the minimization of unpaired cysteines in the humanized antibody sequences 206 can increase the stability of antibodies synthesized based on the humanized antibody sequences 206.

[0052] In one or more illustrative examples, the humanized antibody sequences 206 can include single-domain antibodies. The single-domain antibodies can include a number of framework regions and a number of complementarity determining regions. In at least some examples, the single-domain antibodies can include at least two CDRs, at least three CDRs, at least four CDRs, or at least five CDRs. Additionally, the single-domain antibodies can include at least two framework regions, at least three framework regions, at least four framework regions, at least five framework regions, or at least six framework regions. In one or more examples, the sequence modification data 212 can indicate that at least a portion of the amino acids located in the CDRs of the single-domain antibodies are to be conserved and not modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. In one or more additional example, the sequence modification data 212 can indicate that each of the amino acids located in the CDRs of the single-domain antibodies are to be conserved and not be modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. In various examples, the sequence modification data 212 can indicated that one or more amino acids located in the framework regions of the single-domain antibodies are to be conserved and not modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. Further, the sequence modification date 212 can indicate that amino acids located in the framework regions at one or more positions adjacent to the CDRs are to be conserved and not be modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. In one or more further examples, the sequence modification data 212 can indicate positions of the single-domain antibodies that can be modified by the antibody humanization system 202 to generate the humanized antibody sequences 206. In still additional illustrative examples, the sequence modification data 212 can indicate one or more positions in the framework regions of single-domain antibodies that can be modified by the antibody humanization system 202 to generate the humanized antibody sequences 206.

[0053] In one or more examples, the humanized antibody sequences 206 can be training data that is provided to a generative adversarial network 214. The generative adversarial network 214 can generate amino acid sequences of antibodies or amino acid sequences of antibody segments that correspond to the humanized antibody sequences 206 after being trained. The generative adversarial network 214 can include at least one of generative adversarial network computer-readable instructions, generative adversarial network logic, or generative adversarial network circuitry.

[0054] In one or more illustrative examples, the humanized antibody sequences 206 can include antibody light chains and the generative adversarial network 214 can generate amino acid sequences that correspond to antibody light chains. In one or more additional illustrative examples, the humanized antibody sequences 206 can include antibody heavy chains and the generative adversarial network 214 can generate amino acid sequences that correspond to antibody heavy chains. In one or more further illustrative examples, the humanized antibody sequences 206 can include both antibody heavy chains and antibody light chains and the generative adversarial network 214 can generate amino acid sequences of antibodies that correspond to both antibody heavy chains and antibody light chains. In still further examples, the humanized antibody sequences 206 can include single-domain antibodies and the generative adversarial network 214 can generate amino acid sequences that correspond to single-domain antibodies. In one or more additional examples, the humanized antibody sequences 206 can include VHHs and the generative adversarial network 214 can generate amino acid sequences that correspond to VHHs.

[0055] The generative adversarial network architecture 214 can include a generating component 216 and a challenging component 218. The generating component 216 can implement one or more machine learning computational models to generate amino acid sequences based on input provided to the generating component 216. In various implementations, the one or more machine learning computational models implemented by the generating component 216 can include one or more functions and one or more weights. The challenging component 218 can generate output indicating whether the amino acid sequences produced by the generating component 216 correspond to various characteristics. The output produced by the challenging component 218 can be provided to the generating component 216 and the one or more machine learning computational models implemented by the generating component 216 can be modified based on the feedback provided by the challenging component 218. In various implementations, the challenging component 218 can analyze the amino acid sequences generated by the generating component 216 with amino acid sequences of proteins included in training data and generate an output indicating an amount of correspondence between the amino acid sequences produced by the generating component 216 and the amino acid sequences of proteins provided to the challenging component 218 as training data. In one or more illustrative examples, the analysis performed by the challenging component 218 with respect to the amino acid sequences produced by the generating component 216 can include a comparison between the amino acid sequences included in the training data and the amino acid sequences produced by the generating component 216.

[0056] In various implementations, the generative adversarial network architecture 214 can implement one or more artificial neural network technologies. For example, the generative adversarial network architecture 214 can implement one or more recurrent neural networks. Additionally, the generative adversarial network architecture 214 can implement one or more convolutional neural networks. In one or more implementations, the generative adversarial network architecture 214 can implement a combination of recurrent neural networks and convolutional neural networks. In one or more additional examples, the generating component 216 can include a generator and the challenging component 218 can include a discriminator. In one or more further implementations, the generative adversarial network architecture 214 can include a Wasserstein generative adversarial network (wGAN). In these scenarios, the generating component 216 can include a generator and the classifying component 306 can include a critic.

[0057] In the illustrative example of Figure 2, input data 220 can be provided to the generating component 216 and the generating component 216 can produce one or more generated sequences 222 from the input data 220 using one or more machine learning computational models. In one or more implementations, the input data 220 can include noise data that is generated by a random number generator or a pseudo-random number generator. The generated sequence(s) 222 can be compared by the challenging component 218 against the humanized antibody sequences 206 that have been structured according to one or more schemas.

[0058] Based on similarities and/or differences between the generated sequence(s) 222 and the humanized antibody sequences 206, the challenging component 218 can generate a classification output 224 that indicates an amount of similarity and/or an amount of difference between the generated sequence 222 and the humanized antibody sequences 206. In one or more examples, the challenging component 218 can label the generated sequence(s) 222 as zero and the humanized antibody sequences 206 can be labeled as one. In these situations, the classification output 224 can correspond to a number from 0 and 1. In additional examples, the challenging component 218 can implement a distance function that produces an output that indicates an amount of distance between the generated sequence(s) 222 and humanized antibody sequences 206. In these scenarios, the challenging component 218 can label the generated sequence(s) 222 as -1 and the encoded humanized antibody sequences 206 as 1. In implementations where the challenging component 218 implements a distance function, the classification output 224 can be a number from $-\infty$ to ∞. In various examples, the humanized antibody sequences 206 can be referred to as ground truth data.

[0059] The humanized antibody sequences 206 can be subject to data preprocessing 226 before being provided to the challenging component 218. In one or more implementations, the humanized antibody sequences 206 can be arranged according to a classification system before being provided to the challenging component 218. The data preprocessing 226 can include pairing amino acids included in the antibodies that correspond to the humanized antibody sequences 206 with numerical values that can represent structure-based positions within the antibodies. The numerical values can include a sequence of numbers having a starting point and an ending point. In an illustrative example, a T can be paired with the number 43 indicating that a Threonine molecule is located at a structure-based position 43 of a specified antibody domain type. In one or more illustrative examples, structure-based numbering can be applied to antibodies, antibody fragments, and antibody-like molecules, such as fibronectin type III (FNIII) proteins, avimers, VHH domains, kinases, T-cells, zinc fingers, and the like.

[0060] In one or more implementations, the classification system implemented by the data preprocessing 226 can designate a particular number of positions for certain regions of antibodies. For example, the classification system can designate that portions of antibodies having particular functions and/or characteristics can have a specified number of positions. In various situations, not all of the positions included in the classification system may be associated with an amino acid because the number of amino acids in a specified region of an antibody may vary between antibodies. To illustrate, the number of amino acids in a region of an antibody can vary for different types of antibodies, antibody segments, and/or antibody-like molecules. In one or more examples, positions of the classification system that are not associated with a particular amino acid can indicate various structural features of an antibody, such as a turn or a loop. In an illustrative example, a classification system for the humanized antibody sequences 206 can indicate that at least one of heavy chain regions, light chain regions, framework regions, antigen binding regions, CDRs, framework regions, or hinge regions have a specified number of positions assigned to them and the amino acids of the antibodies can be assigned to the positions according to the classification system.

[0061] The output produced by the data preprocessing 226 can include structured sequences 228. The structured sequences 228 can include a matrix indicating amino acids associated with various positions of an antibody, antibody segment, or antibody-like molecule. In one or more examples, the structured sequences 228 can include a matrix having columns corresponding to different amino acids and rows that correspond to structure-based positions of antibodies. For each element in the matrix, a 0 can be used to indicate the absence of an amino acid at the corresponding position and a 1 can be used to indicate the presence of an amino acid at the corresponding position. In situations where a position represents a gap in an amino acid sequence, the row associated with the position can comprise zeroes for each column. The generated sequence(s) 222 can also be represented using a vector according to a same or similar number scheme as used for the structured sequences 228. In one or more illustrative examples, the structured sequences 228 and the generated sequence(s) 222 can be encoded using a method that may be referred to as a one-hot encoding method.

[0062] In one or more examples, the training process for the generative adversarial network architecture 214 can be complete after the function(s) implemented by the generating component 216 and the function(s) implemented by the challenging component 218 converge.

The convergence of a function can be based on the movement of values of model parameters toward specified values as antibody sequences are generated by the generating component 216 and feedback is obtained from the challenging component 218. In various implementations, the training of the generative adversarial network architecture 214 can be complete when the antibody sequences generated by the generating component 216 have one or more specified characteristics. To illustrate, the amino acid sequences generated by the generating component 216 can be analyzed by a software tool that can analyze amino acid sequences to determine at least one of biophysical properties of the amino acid sequences, structural features of the amino acid sequences, or adherence to amino acid sequences corresponding to one or more antibody germlines.

[0063] After the generative adversarial network architecture 214 has undergone a training process, the trained generating component 216 can produce humanized antibody sequence data 230. The humanized antibody sequence data 230 can include amino acid sequences of antibodies, antibody segments, single-domain antibodies, antibody light chains, antibody heavy chains, T-cells receptors, affibodies, or one or more combinations thereof. In various examples, the amino acid sequences included in the humanized antibody sequence data 230 can be used to generate a library of antibody sequences. For example, the humanized antibody sequence data 230 can include from about 10,000 amino acid sequences to about 100,000 amino acid sequences that can be used to generate a library of antibody sequences that includes from about 50,000 amino acid sequences to about 1,000,000 amino acid sequences, from about 100,000 amino acid sequences to about 1,000,000 amino acid sequences, from about 100,000 amino acid sequences to about 500,000 amino acid sequences, or from about 500,000 amino acid sequences to about 1,000,000 amino acid sequences within a given duration, such as no greater than hour, no greater than three hours, no greater than six hours, no greater than ten hours, or no greater than twelve hours.

[0064] In one or more illustrative examples, the protein sequence generating system 114 can include a trained version of the generating component 216 and the humanized antibody sequence data 230 can be provided to the protein sequence fragmentation system 118. In these examples, the protein sequence fragmentation system 118 can divide individual antibody sequences included in the humanized antibody sequence data 230 into multiple fragments. The protein sequence assembly system 124 can then reconstruct a number of antibody sequences using the fragments generated by the protein sequence fragmentation system 118 based on the amino acid sequences included in the humanized antibody sequence data 230. The reconstructed antibody sequences can then be analyzed by the sequence analysis system 128. In various examples, the sequence analysis system 128 can determine whether the reconstructed antibody sequences correspond to the antibody sequences included in the humanized antibody sequence data 230. In one or more examples, the sequence analysis system 128 can determine whether the reconstructed antibody sequences have one or more characteristics of the amino acid sequences included in the humanized antibody sequence data 230 produced by the generative adversarial network architecture 214.

[0065] Figure 3 is a diagram illustrating an example framework 300 to perform transfer learning with respect to a generative adversarial network architecture 302, in accordance with one or more implementations. By implementing transfer learning techniques with respect to the generative adversarial network architecture 302, amino acid sequences of proteins can be generated that have one or more specified structural features and/or one or more specified biophysical properties. In one or more illustrative examples, the generative adversarial network architecture 302 can include the generative adversarial network architecture 214 described with respect to Figure 2.

[0066] The generative adversarial network architecture 302 can include a generating component 304 and a challenging component 306. The generating component 304 can implement one or more models to generate amino acid sequences based on input provided to the generating component 304. In various implementations, the one or more models implemented by the generating component 304 can include one or more functions. The challenging component 306 can generate output indicating whether the amino acid sequences produced by the generating component 304 correspond to various characteristics. The output produced by the challenging component 306 can be provided to the generating component 304 and the one or more models implemented by the generating component 304 can be modified based on the feedback provided by the challenging component 306. In various implementations, the challenging component 306 can analyze the amino acid sequences generated by the generating component 304 with amino acid sequences of proteins included in training data and generate an output indicating an amount of correspondence between the amino acid sequences produced by the generating component 304 and the amino acid sequences of proteins provided to the challenging component 306 as training data. In one or more illustrative examples, the analysis performed by the challenging component 306 with respect to the amino acid sequences produced by the generating component 304 can include a comparison between the amino acid sequences included in the training data and the amino acid sequences produced by the generating component 304.

[0067] In various implementations, the generative adversarial network architecture 302 can implement one or more artificial neural network technologies. For example, the generative adversarial network architecture 302 can implement one or more recurrent neural networks. Additionally, the generative adversarial network architecture 302 can implement one or more convolutional neural networks. In one or more implementations, the generative adversarial network architecture 302 can implement a combination of recurrent neural networks and convolutional neural networks. In one or more additional examples, the generating component 304 can include a generator and the challenging component 306 can include a discriminator. In one or more further implementations, the generative adversarial network architecture 302 can include a Wasserstein generative adversarial network (wGAN). In these scenarios, the generating component 304 can include a generator and the challenging component 306 can include a critic.

[0068] In the illustrative example of Figure 3, first input data 308 can include an input vector that is provided to the generating component 304 and the generating component 304 can produce one or more generated sequences 310 from the first input data 308 using one or more models. In one or more implementations, the first input data 308 can include noise data that is generated by a random number generator or a pseudo-random number generator. The generated sequence(s) 310 can be compared by the challenging component 306 against sequences of proteins included in first protein sequence data 312 that have been structured according to one or more schemas. The first protein sequence data 312 can include sequences of proteins obtained from one or more data sources that store protein sequences. The first protein sequence data 312 can be training data for the generative adversarial network architecture 302. In one or more illustrative examples, the first protein sequence data 312 can include the humanized antibody sequences 206 described with respect to Figure 2.

[0069] Based on similarities and/or differences between the generated sequence(s) 310 and the sequences obtained from the first protein sequence data 312, the challenging component 306 can generate a classification output 314 that indicates an amount of similarity and/or an amount of difference between the generated sequence 310 and sequences included in the first protein sequence data 312. In one or more examples, the challenging component 306 can label the generated sequence(s) 310 as zero and the sequences obtained from the first protein sequence data 312 as one. In these situations, the classification output 314 can correspond to a number from 0 and 1. In additional examples, the challenging component 306 can implement a distance function that produces an output that indicates an amount of distance between the generated sequence(s) 310 and the proteins included in the first protein sequence data 312. In these scenarios, the challenging component 306 can label the generated sequence(s) 310 as -1 and the encoded amino acid sequences obtained from the first protein sequence data 312 as 1. In implementations where the challenging component 306 implements a distance function, the classification output 314 can be a number from $-\infty$ to ∞. In at least some examples, the amino acid sequences obtained from the first protein sequence data 312 can be referred to as ground truth data.

[0070] The protein sequences included in the first protein sequence data 312 can be subject to data preprocessing 316 before being provided to the challenging component 306. In one or more implementations, the first protein sequence data 312 can be arranged according to a classification system, also referred to as a classification schema, before being provided to the challenging component 306. The data preprocessing 316 can include pairing amino acids included in the proteins of the first protein sequence data 312 with numerical values that can represent structure-based positions within the proteins. In one or more illustrative examples, structure-based numbering can be applied to any general protein type, such as fibronectin type III (FNIII) proteins, avimers, antibodies, antibody segments, antibody light chains, antibody heavy chains, VHH domains, kinases, zinc fingers, and the like.

[0071] In one or more implementations, the classification system implemented by the data preprocessing 316 can designate a particular number of positions for certain regions of proteins. For example, the classification system can designate that portions of proteins have particular functions and/or characteristics can have a specified number of positions. In various situations, not all of the positions included in the classification system may be associated with an amino acid because the number of amino acids in a specified region of a protein may vary between proteins. To illustrate, the number of amino acids in a region of a protein can vary for different types of proteins. In one or more examples, positions of the classification system that are not associated with a particular amino acid can indicate various structural features of a protein, such as a turn or a loop. In an illustrative example, a classification system for antibodies can indicate that heavy chain regions, light chain regions, framework regions, CDRs, antigen binding regions, and/or hinge regions have a specified number of positions assigned to them and the amino acids of the antibodies can be assigned to the positions according to the classification system.

[0072] The data used to train the generative adversarial network architecture 302 can impact the amino acid sequences produced by the generating component 304. For example, in situations where antibodies are included in the first protein sequence data 312 provided to the challenging component 306, the amino acid sequences generated by the generating component 304 can correspond to antibody amino acid sequences. In one or more additional examples, in scenarios where T-cell receptors are included in the first protein sequence data 312 provided to the challenging component 306 the amino acid sequences generated by the generating component 304 can correspond to T-cell receptor amino acid sequences. In one or more additional examples, in situations where kinases are included in the first protein sequence data 312 provided to the challenging component 306, the amino acid sequences generated by the generating component 304 can correspond to amino acid sequences of kinases. In implementations where amino acid sequences of a variety of different types of proteins are included in the first protein sequence data 312 provided to the challenging component 306, the generating component 304 can generate amino acid sequences having characteristics of proteins generally and may not correspond to a particular type of protein. Further, in various examples, the amino acid sequences produced by the generating component 304 can correspond to the types of proportions of amino acid sequences included in the first protein sequence data 312 provided to the challenging component 306.

[0073] The output produced by the data preprocessing 316 can include structured sequences 318. The structured sequences 318 can include a matrix indicating amino acids associated with various positions of a protein. In one or more examples, the structured sequences 318 can include a matrix having columns corresponding to different amino acids and rows that correspond to structure-based positions of proteins. For each element in the matrix, a 0 can be used to indicate the absence of an amino acid at the corresponding position and a 1 can be used to indicate the presence of an amino acid at the corresponding position. In situations where a position represents a gap in an amino acid sequence, the row associated with the position can comprise zeroes for each column. The generated sequence(s) 310 can also be represented using a vector according to a same or similar number scheme as used for the structured sequences 318. In at least some illustrative examples, the structured sequences 318 and the generated sequence(s) 310 can be encoded using a method that may be referred to as a one-hot encoding method.

[0074] After the generative adversarial network architecture 302 has undergone a training process, one or more first trained generating components 320 can be generated that correspond to a trained version of at least the generating component 304 that can produce amino acid sequences of proteins. In various examples, the generative adversarial network architecture 302 can include multiple generating components. For example, the generative adversarial network architecture 302 can include a first generating component that generates amino acid sequences of a first number of proteins or protein fragments and a second generating component that generates amino acid sequences of a second number of proteins or protein fragments. In one or more illustrative examples, the first generating component can generate amino acid sequences of antibody light chains and the second generating component can generate amino acid sequences of antibody heavy chains that can be combined to produce a single antibody amino acid sequence.

[0075] In one or more examples, the training process for the generative adversarial network architecture 302 can be complete after the function(s) implemented by the generating component 304 and the function(s) implemented by the challenging component 306 converge. The convergence of a function can be based on the movement of values of model parameters toward specified values as protein sequences are generated by the generating component 304 and feedback is obtained from the challenging component 306. In various implementations, the training of the generative adversarial network architecture 302 can be complete when the protein sequences generated by the generating component 304 have one or more specified characteristics. To illustrate, the amino acid sequences generated by the generating component 304 can be analyzed by a software tool that can analyze amino acid sequences to determine at least one of biophysical properties of the amino acid sequences, structural features of the amino acid sequences, or adherence to amino acid sequences corresponding to one or more protein germlines.

[0076] The first trained generating components 320 can undergo transfer learning at 322 based on second protein sequence data 324. The transfer learning that is performed at 322 can modify the one or more first trained generating components 320 based on the amino acid sequences included in the second protein sequence data 324. The transfer learning that is performed at 322 can be performed using one or more modified generative adversarial network architectures 326. The one or more modified generative adversarial network architectures 326 can include at least a portion of the one or more first trained generating components 320 and one or more additional challenging components. In various examples, the transfer learning that takes place at 322 can include an additional training process of the one or more first trained generating components 320 using training data obtained from the second protein sequence data 324. By using a training dataset to produce the one or more second trained generating components 328 that is different from the training dataset used to produce the one or more first trained generating components 320, the one or more modified generative adversarial network architectures 326 can produce amino acid sequences that can have some general characteristics that correspond to the amino acid sequences included in the first protein sequence data 312 and that also have one or more specified characteristics that correspond to features of the proteins related to the amino acid sequences included in the second protein sequence data 324.

[0077] In various implementations, the one or more first trained generating components 320 can be further trained using the second protein sequence data 324 as part of the transfer learning at 322 to produce one or more second trained generating components 328 in a manner that is similar to the training of the generative adversarial network architecture 302 that produced the one or more first trained generating components 320. In one or more examples, components of the one or more modified generative adversarial network architectures 326 can be trained to minimize at least one loss function. Additionally, the training process used in the transfer learning at 322 to produce the one or more second trained generating components 328 can be complete after one or more modified functions implemented by the one or more modified generative adversarial network architectures 326 converge. In one or more further examples, the training process used to generate the one or more second trained generating components 328 from the one or more first trained generating components 320 can be complete based on an analysis of a software tool indicating that amino acid sequences produced using the one or more modified generative adversarial network architectures 326 corresponds to one or more specified criteria. The one or more specified criteria can correspond to at least one of one or more structural features of proteins or one or more biophysical properties of proteins. In still other examples, the first trained generating components 320 can be further trained using the second protein sequence data 324 to at least one of increase stability of proteins in one or more environments, to increase binding of proteins to one or more target molecules, or to increase the probability of proteins being viable after one or more manufacturing processes.

[0078] In one or more examples, the second protein sequence data 324 can include amino acid sequences of proteins that have features that are different from the features of the proteins related to the first protein sequence data 312. In various examples, the second protein sequence data 324 can include a subset of the amino acid sequences included in the first protein sequence data 312. In one or more additional examples, the second protein sequence data 324 can include a greater number of a group of amino acid sequences having one or more specified characteristics in relation to the number of amino acid sequences having the one or more characteristics included in the first protein sequence data 312. For example, the first protein sequence data 312 can include amino acid sequences of proteins having a variety of structural features. To illustrate, the first protein sequence data 312 can include a number of amino acid sequences of proteins having one or more sizes of hydrophobic regions, a number of amino acid sequences of proteins having one or more sizes of negatively charged regions, a number of amino acid sequences of proteins having one or more sizes of positively charged regions, a number of amino acid sequences of proteins one or more sizes of polar regions, one or more combinations thereof, and the like. In one or more implementations, the second protein sequence data 324 can include amino acid sequences of proteins that have a greater number of amino acid sequences of proteins having a subset of the properties of the proteins included in the first protein sequence data 312, such as a greater number of amino acid sequences of proteins that have hydrophobic regions with a specified range of sizes than the number of amino acid sequences included in the first protein sequence data 312 that have the hydrophobic regions with the specified range of sizes. In these scenarios, the one or more second trained generating components 328 can primarily produce amino acid sequences of proteins having hydrophobic regions with the specified range of sizes.

[0079] In one or more implementations, the amino acid sequences included in the second protein sequence data 324 can include a filtered set of amino acid sequences. For example, a set of amino acid sequences can be evaluated according to one or more criteria. In various examples, at least one of one or more software tools, one or more diagnostic tools, or one or more analytical instruments can be used to identify amino acid sequences included in the set of amino acid sequences that correspond to the one or more criteria. The amino acid sequences that satisfy the one or more criteria can then be added to the second protein sequence data 324. In one or more illustrative examples, a number of amino acid sequences can be evaluated to identify proteins having at least one polar region for inclusion in the second protein sequence data 324. In these scenarios, the amino acid sequences that include at least one polar region can be used to modify the one or more first trained generating components 320 during the transfer learning at 322 to produce the one or more second trained generating components 328 that can produce amino acid sequences of proteins having at least one polar region.

[0080] The one or more second trained generating components 328 can generate amino acid sequences based on second input data 330. In one or more examples, the second input data 330 can include a random or pseudo-random series of numbers that can be used by the one or more second trained generating components 328 to produce amino acid sequences. In various examples, the one or more second trained generating components 328 can include multiple generating components that each generate amino acid sequences of proteins having a different distribution of values for a structural feature. For example, the second trained generating components 328 can include a first generating component that produces amino acid sequences of proteins having one or more negatively charged regions with a first range of sizes and a second generating component that produces amino acid sequences of proteins having one or more negatively charged regions with a second range of sizes. In one or more examples, there may be some overlap between the sizes of negatively charged regions included in the first range of sizes and the second range of sizes.

[0081] The one or more second trained generating components 328 can generate additional protein sequences 332. The additional protein sequences 332 can correspond to amino acid sequences of proteins having characteristics that correspond to characteristics of proteins that correspond to the amino acid sequences included in the second protein sequence data 324. In one or more examples, the additional protein sequences 332 can include amino acid sequences of proteins having at least one of one or more structural features or values of biophysical properties of proteins that correspond to the amino acid sequences included in the second protein sequence data 324. That is, by performing an additional training process with respect to the one or more first trained generating components 320 at the transfer learning process of 322 using training data that corresponds to proteins have one or more structural features of interest and/or having specified values of one or more biophysical properties of interest, the proteins that correspond to the amino acid sequences of the additional protein sequences 332 can also have at least a threshold probability of having the one or more structural features of interest and/or the specified values of the one or more biophysical properties of interest. Thus, the framework 300 can be implemented in scenarios where a library of proteins can be produced having one or more structural features of interest and/or specified values of one or more biophysical properties of interest. Additionally, by leveraging the learning that takes place to produce the one or more first trained generating components 320 followed by the transfer learning at 322 using a more specialized training dataset, the computing resources used to generate the additional protein sequences 332 can be minimized and the accuracy of the characteristics of interest for the proteins corresponding to the amino acid sequences included in the additional protein sequences 332 can be increased in relation to previous techniques.

[0082] Further, although a single transfer learning process at 322 is described with respect to the illustrative example of Figure 3, multiple additional transfer learning processes for the generative adversarial networks can be performed. In one or more examples, multiple structural features and/or multiple biophysical properties can be of interest with respect to a library of proteins. In these scenarios, an additional training dataset that includes amino acid sequences of one or more of the structural features and/or biophysical properties of interest can be used in one or more additional transfer learning processes to further train the generating components of the generative adversarial networks. Thus, with each additional training process and subsequent modifications to the computational layers of the generative adversarial network generating components, the characteristics of the proteins corresponding to the amino acid sequences generated by the trained generative adversarial networks can be further modified.

[0083] The additional protein sequences 332 can be provided to the protein sequence fragmentation system 118 and the protein sequence fragmentation system 118 can produce the protein sequence fragments 120. In these scenarios, the second trained generating components 328 can be included in the protein sequence generating system 114 described with respect to Figure 1. In one or more illustrative examples, the second trained generating component 328 can produce amino acid sequences of at least one of antibodies or antibody segments having at least one of structural features or biophysical properties that correspond to at least one of antibodies or antibody fragments included in the second protein sequence data 324. Continuing with this example, the protein sequence fragmentation system 118 can generate the protein sequence fragments 120 that include the amino acid sequences of the antibodies and/or antibody segments produced by the one or more second trained generating components 328. The fragments of the antibodies and/or antibody segments can then be reassembled by the protein sequence assembly system 124 to generate the reconstructed protein sequences 126. In these scenarios, the reconstructed proteins sequences 126 can include amino acid sequences of at least one of antibodies or antibody segments that are to be evaluated by the sequence analysis system 128 to determine a group of the reconstructed amino acid sequences to include in a library of amino acid sequences that correspond to antibodies and/or antibody segments.

[0084] Figure 4 is a diagram illustrating an example framework 400 to divide amino acid sequences of single-domain antibodies into a number of fragments and re-assemble the fragments, in accordance with one or more implementations. In one or more examples, the single-domain antibodies can include humanized VHHs that correspond to VHHs originally produced by one or more camelids.

[0085] The framework 400 can include an amino acid sequence 402 of a single-domain antibody. The amino acid sequence 402 can include a number of regions. In various examples, the amino acid sequence 402 can include a number of discrete regions that have one or more characteristics and/or one or more functions. In one or more additional examples, the discrete regions of the amino acid sequence 402 can include a specified number of positions of amino acids. In at least some examples, each position of a given region can be occupied by an amino acid, while in other situations, at least one position of a given region can be unoccupied and devoid of an amino acid. In one or more examples, a number of positions that comprise individual regions of the amino acid sequence 402 can be predetermined. To illustrate, a number of positions that comprised individual regions of the amino acid sequence 402 can be defined by a scheme that designates groups of positions of an amino acid sequence as corresponding to one or more regions.

[0086] The amino acid sequence 402 includes a number of framework regions and a number of CDRs. In various examples, the CDRs can also be referred to herein as variable regions. The individual framework regions can have amino acid sequences that are more likely to be preserved across different antibodies than the amino acid sequences of the CDRs. For example, antibodies and antibody segments that bind to different antigens are more likely to have differences in the amino acid sequences of the CDRs than in the framework regions. In the illustrative example of Figure 4, the amino acid sequence 402 includes a first framework region 404, a second framework region 406, a third framework region 408, and a fourth framework region 410. The amino acid sequence 402 can also include a first CDR 412, a second CDR 414, and a third CDR 414.

[0087] In at least some examples, a number of positions included in the individual framework regions 404, 406, 408, 410 can be different from one another. To illustrate, the first framework region 404 can include a first number of positions, the second framework region 406 can include a second number of positions, the third framework region 408 can include a third number of positions, and the fourth framework region 410 can include a fourth number of positions. In one or more examples, at least one of the first number of positions, the second number of positions, the third number of positions, or the fourth number of positions can be different from another one of the first number of positions, the second number of positions, the third number of positions, or the fourth number of positions. Additionally, in various examples, the number of positions included in CDRs can be different from one another. For example, the first CDR 412 can include a first additional number of positions, the second CDR 414 can include a second additional number of positions, and the third CDR 416 can include a third additional number of positions. In one or more examples, at least one of the first additional number of positions, the second additional number of positions, or the third additional number of positions can be different from another one of the first additional number of positions, the second additional number of positions, or the third additional number of positions.

[0088] In one or more examples, the first framework region 404 can include from 28 to 35 positions, the second framework region 406 can include from 8 to 18 positions, the third framework region 408 can include from 25 to 36 positions, and the fourth framework region 410 can include from 7 to 15 positions. Further, the first CDR 412 can include from 6 to 14 positions, the second CDR 414 can include from 15 to 25 positions, and the third CDR 416 can include from 25 to 35 positions. In various examples, the amino acid sequence 402 can be represented by 140 to 155 positions. In one or more illustrative examples, the amino acid sequence 402 can be represented by 149 positions with the first framework region 404 corresponding to 32 positions, the first CDR 412 corresponding to 10 positions, the second framework region 406 corresponding to 14 positions, the second CDR 414 corresponding to 20 positions, the third framework region 408 corresponding to 32 positions, the third CDR 416 corresponding to 30 positions, and the fourth framework region 410 corresponding to 11 positions.

[0089] The framework 400 can include, at operation 418, generating antibody sequence fragments. The antibody sequence fragments can include segments of the amino acid sequence 402. In one or more examples, the antibody sequence fragments can be generated according to one or more computational techniques or methods. For example, the antibody sequence fragments can be generated using at least one of one or more machine learning techniques or one or more statistical techniques. In various examples, the antibody sequence fragments can be generated according to one or more schema or rules. In at least some examples, at least a portion of the antibody sequence fragments generated at operation 418 can have overlapping positions. In one or more additional examples, the antibody sequence fragments generated at operation 418 can be produced such that the positions of at least one region of the amino acid sequence 402 are continuous and not separated. In one or more illustrative examples, the operations described with respect to operation 418 can be performed, at least in part, by the protein sequence fragmentation system 118 described with respect to Figure 1.

[0090] In the illustrative example of Figure 4, the amino acid sequence 402 can be divided into three segments at operation 418. In one or more examples, the amino acid sequence 402 can be divided into fragments that each include a separate CDR region and that have overlapping portions with a framework region of at least one other fragment. For example, at operation 418, a first antibody sequence fragment 420 can be generated that spans from a first position 422 to a second position 424. In various examples, the first antibody sequence fragment 420 can be generated by cleaving the amino acid sequence 402 at the second position 424. In these scenarios, the first antibody sequence fragment 420 can include the first framework region 404, the first CDR 412, and a portion of the second framework region 406.

[0091] Additionally, at operation 418, a second antibody sequence fragment 426 can be generated that spans from a third position 428 to a fourth position 430. In various examples, the second antibody sequence fragment 426 can be generated by cleaving the amino acid sequence 402 at the third position 428 and the fourth position 430. In these situations, the second antibody sequence fragment 426 can include the second framework region 406, the second CDR 414, and a portion of the third framework region 408. The first antibody sequence fragment 420 and the second antibody sequence fragment 426 can overlap in the positions of the second framework region 406 indicated by the first overlap region 432.

[0092] Further, at operation 418, a third antibody sequence fragment 434 can be generated that spans from a fifth position 436 to a sixth position 438. In one or more examples, the third antibody sequence fragment 434 can be generated by cleaving the amino acid sequence 402 at the fifth position 436. In these instances, the third antibody sequence fragment 434 can include a portion of the third framework region 408, the third CDR 416, and the fourth framework region 410. The second antibody sequence fragment 426 can overlap the third antibody sequence fragment 434 in the positions of the third framework region 408 indicated by the second overlap region 440. In the illustrative example of Figure 4, the antibody sequence fragments 420, 426, 434 each include an entire CDR region, an entire framework region, and a partial framework region that overlaps with a framework region included in another antibody sequence fragment.

[0093] In various examples, the positions where the amino acid sequence 402 is cleaved to generate the antibody sequence fragments 420, 426, 434 can be determined based on one or more predetermined schema or rules. For example, the positions where the amino acid sequence 402 is cleaved can be determined based on one or more schema or rules indicating that individual antibody fragments are to include at least one CDR and at least a portion of each framework region bordering the CDR. In one or more additional examples, the one or more schema or rules used to cleave the amino acid sequence 402 at operation 418 can assign a number to each position of the amino acid sequence 402 in ascending order starting from the first position 422 to the sixth position 438. The one or more schema or rules used to cleave the amino acid sequence 402 to generate the antibody sequence fragments 420, 426, 434 can also indicate that each antibody sequence fragment includes at least one entire framework region and one entire CDR.

[0094] Further, the one or more schema or rules used to generate the antibody sequence fragments can indicate a position within the framework regions at which to cleave the respective framework regions. To illustrate, the one or more schema or rules used to generate the antibody sequence fragments 420, 426, 434 can indicate that the amino acid sequence 402 is to be cleaved at the second position 424, the fourth position 430, and the fifth position 436. In one or more further examples, the one or more rules or schema used to generate the antibody sequence fragments 420, 426, 434 can indicate a range of positions within the second framework region 406 and the third framework region 408 where the amino acid sequence 402 can be cleaved. In one or more illustrative examples, the individual positions and/or ranges of positions included in the one or more schema or rules used to generate the antibody sequence fragments 420, 426, 434 can be determined based on an analysis of other antibody sequences using at least one of one or more statistical techniques or one or more machine learning techniques. For example, the individual positions and/or ranges of positions included in the one or more schema or rules used to generate the antibody sequence fragments 420, 426, 434 can be determined based on an analysis of least one of the analysis of reconstructed sequences produced using the antibody sequence fragments 420, 426, 434 or the analysis of amino acid sequences of antibodies that have a same or similar function to that of the amino acid sequence 402. In at least some examples, the amino acid sequence 402 can be cleaved into more than three fragments. In addition, the amino acid sequence 402 can undergo the process performed at operation 418 multiple times to produce a number of different sets of antibody sequence fragments.

[0095] Although the illustrative example of Figure 4 shows using the amino acid sequence 402 to produce three antibody sequence fragments, in one or more other implementations, the amino acid sequence 402 can be divided into fewer antibody sequence fragment or into a greater number of antibody sequence fragments. Additionally, the operation 418 can be performed with respect to a number of antibody sequences to produce thousands of antibody sequence fragments, up to tens of thousands of antibody sequence fragments, up to hundreds of thousands of antibody sequence fragments, or more.

[0096] In one or more examples, the antibody sequence fragments can be grouped into a number of pools of antibody sequence fragments. Individual pools of the antibody sequence fragments can include antibody sequence fragments that correspond to at least one of one or more CDRs or one or more framework regions. In one or more illustrative examples, the antibody sequence fragments can be grouped into a first pool 442 that includes first antibody sequence fragments having amino acid sequences that correspond to the first CDR 412, a second pool 444 that includes second antibody sequence fragments having amino acid sequences that correspond to the second CDR 414, and a third pool 446 that includes third antibody sequence fragments having amino acid sequences that correspond to the third CDR 416.

[0097] In various examples, the antibody sequence fragments included in each pool can also include amino acid sequences of one or more framework regions. For example, antibody sequence fragments included in the first pool 442 can include amino acid sequences of at least a portion of the first framework region 404 and at least a portion of the second framework region 406 in addition to amino acid sequences of the first CDR 412. In addition, antibody sequence fragments included in the second pool 444 can include amino acid sequences of at least a portion of the second framework region 406 and at least a portion of the third framework region 408 in addition to amino acid sequences of the second CDR 414. Further, antibody sequence fragments included in the third pool 446 can include amino acid sequences of at least a portion of the third framework region 408 and at least a portion of the fourth framework region 410 in addition to amino acid sequences of the third CDR 416.

[0098] In various examples, individual antibody sequence fragments may include and/or be associated with metadata indicating a pool that corresponds to the individual antibody sequence fragments. To illustrate, the first pool 442 can correspond to a first identifier, the second pool 444 can correspond to a second identifier, and the third pool 446 can correspond to a third identifier. A given identifier can be assigned to an individual antibody sequence fragment at operation 418. The antibody sequence fragments generated in conjunction with operation 418 can be grouped into pools based on the identifiers associated with the individual antibody sequence fragments. In one or more examples, the identifiers can be determined for antibody sequence fragments based on a set of positions within a scheme that indicates a total number of positions of an antibody sequence and that indicates respective group of positions that correspond to individual framework regions and/or individual CDRs. In at least some examples, the positions of the scheme can be associated with individual amino acid sequences

402 as at least a portion of the metadata that corresponds to individual amino acid sequences 402. In this way, when the amino acid sequences 402 are divided into antibody sequence fragments, the positions of the scheme that correspond to the individual antibody sequence fragments are indicated in relation to individual antibody sequence fragments. Antibody sequence fragments can then be reconstructed based on the positions of the scheme that correspond to the individual antibody sequence fragments that comprise a reconstructed antibody sequence.

[0099] In one or more examples, the first pool 442 can correspond to a first number of positions of the scheme, the second pool 444 can correspond to a second number of positions of the scheme, and the third pool 446 can correspond to a third number of positions of the scheme. In one or more illustrative examples, at least one of a start position or an end position for the positions corresponding to the individual pools can vary among antibody sequence fragments. For example, in situations where the scheme includes 149 positions, the first pool 442 can include antibody sequence fragments having a start position of 1 and an end position between 43 and 50, the second pool 444 can include antibody sequence fragments having a start position from 55 to 60, such as position 57, and an end position from 73 to 80, and the third pool 446 can include antibody sequence fragments having a start position from 70 to 75, such as position 73, and an end position of 149.

[00100] In one or more additional examples, a pool that corresponds to an antibody sequence fragment can also be determined by analyzing the amino acids present at a number of positions of the antibody sequence fragments. To illustrate, the first CDR 412 can correspond to a first arrangement of amino acids at a plurality of first positions and the antibody sequence fragments can be analyzed to identify a first group of antibody sequence fragments, such as the first antibody sequence fragment 420, that correspond to the first arrangement. In addition, the second CDR 414 can correspond to a second arrangement of amino acids at a plurality of second positions and the antibody sequence fragments can be analyzed to identify a second group of antibody sequence fragments, such as the second antibody sequence fragment 426 that correspond to the second arrangement. In one or more further examples, the third CDR can correspond to a third arrangement of amino acids at a plurality of third positions and the antibody sequence fragments can be analyzed to identify a third group of antibody sequence fragments, such as the third antibody sequence fragment 434, that correspond to the third arrangement. In these scenarios, the first group of antibody sequence fragments can be included in the first pool 442, the second group of antibody sequence fragments can be included in the second pool 444, and the third group of antibody sequence fragments can be included in the third pool 446. In one or more illustrative examples, the first arrangement, the second arrangement, and the third arrangement can be determined based on an analysis of a number of amino acid sequences that include the first CDR 412, the second CDR 414, and the third CDR 416. In at least some instances, the analysis can be performed using at least one of one or more machine learning techniques or one or more statistical techniques.

[00101] The framework 400 can also include, at operation 448, assembling the antibody sequence fragments included in the first pool 442, the second pool 444, and the third pool 446 into reconstructed antibody sequences 450. The reconstructed antibody sequences 450 can be comprised of an antibody sequence fragment from each of the first pool 442, the second pool 444, and the third pool 446. For example, the reconstructed antibody sequences 450 can be comprised of first antibody sequence components 452 from the first pool 442, second antibody sequence components 454 from the second pool 444, and third antibody sequence components 456 from the third pool 446. In various examples, different reconstructed antibody sequences 450 can include the first antibody sequence fragment 420, the second antibody sequence fragment 426, and the third antibody sequence fragment 434 in conjunction with other antibody sequence fragments generated using additional amino acid sequences included in the first pool 442, the second pool 444, and the third pool 446.

[00102] In one or more examples, the reconstructed antibody sequences 450 can be generated according to one or more schema and/or one or more rules. For example, the reconstructed antibody sequences 450 can be generated such that individual reconstructed antibody sequences 450 include a first antibody sequence component 452 from the first pool 442 that includes the first CDR 412, a second antibody sequence component 454 from the second pool 444 that includes the second CDR 414, and a third antibody sequence component 456 from the third pool 446 that includes the third CDR 416. The first CDR 412, the second CDR 414, and the third CDR 416 can be originally located in different amino acid sequences. In this way, an individual reconstructed antibody sequence 450 can include the first CDR 412 from a first amino acid sequence, the second CDR 414 from a second amino acid sequence, and the third CDR 416 from a third amino acid sequence where the second amino acid sequence is different from the first amino acid sequence and the third amino acid sequence is different from the first amino acid sequence and the second amino acid sequence. Additionally, the reconstructed antibody sequences 450 can be generated such that individual reconstructed antibody sequences 450 include at least one of the first framework region 404, the second framework region 406, the third framework region 408, or a fourth framework region 410 in addition to at least one of the first CDR 412, the second CDR 414, and the third CDR 416.

[00103] In various examples, the amino acids included in the regions of overlap between antibody sequence fragments having different CDR regions can be the same. That is, the amino acids included in the overlap regions 432 and 440 can be conserved between the different amino acid sequences used to generate the antibody sequence fragments included in the pools 442, 444, 446. For example, a first antibody sequence component 452 included in the first pool 442 and having the first CDR 412 can include an amino acid sequence that includes the same amino acids as those located in the first overlap region 432. In these scenarios, the second antibody sequence fragment 426 can be combined with the first antibody sequence component 452 such that the amino acids included in the first overlap region 432 are conserved in a reconstructed antibody sequence 450 that includes the first antibody sequence component 452 and the second antibody sequence fragment 426. In one or more additional examples, first antibody sequence component 452 having the first CDR 412 can include an amino acid sequence that is different from the amino acids located in the first overlap region 432. In these situations, the first antibody sequence component 452 can be combined with the second antibody sequence fragment 426 such that either the amino acid sequence of the first antibody sequence component 452 corresponding to the first overlap region 432 is conserved or the amino acid sequence of the second antibody sequence fragment 426 that corresponds to the first overlap region 432 is conserved in generating a reconstructed antibody sequence 450 that includes the first antibody sequence component 452 and the second antibody sequence fragment 426. In one or more additional examples, a combination of amino acids of the first antibody sequence component 452 and amino acids of the second antibody sequence fragment 426 that correspond to the first overlap region 432 are used to generate a reconstructed antibody sequence 450 in a manner that preserves the number of positions that correspond to the first overlap region 432. In at least some examples, the first antibody sequence component 452 and the second antibody sequence fragment 426 can be combined such that the number of unpaired cysteines in the reconstructed antibody sequence 450 that includes the first antibody sequence component 452 and the second antibody sequence fragment 426. In at least some examples, antibody sequence fragments that do not have the amino acid sequences of the framework regions conserved in the overlap regions 432, 440 can be filtered out and not included in the first pool 442, the second pool 444, or the third pool 446.

[00104] Figure 5 is a diagram of a framework 500 to analyze reconstructed protein sequences for consistency with protein sequences generated by a trained generative machine learning architecture, in accordance with one or more implementations. The framework 500 can include a trained version of the generative machine learning architecture 102. For example, the generative machine learning architecture 102 can include the trained generative machine learning component 110. In one or more illustrative examples, the trained generative machine learning component 110 can include a generating component of a generative adversarial network that has been trained using the protein sequence data 104 described with respect to Figure 1. In various examples, the trained generative machine learning component 110 can include a generating component of a generative adversarial network trained using amino acid sequences of single-domain antibodies.

[00105] The framework 500 can also include the protein sequence assembly system 112. The protein sequence assembly system 112 can generate reconstructed protein sequences 126. The reconstructed protein sequences 126 can include amino acid sequences of proteins that have been generated using amino acid sequence fragments produced by cleaving, using one or more computational or machine learning algorithms, amino acid sequences generated by the trained generative machine learning component 110. In one or more illustrative examples, the reconstructed protein sequences 126 can be generated according to at least one of one or more schema or one or more rules such that the reconstructed protein sequences 126 include one or more specified regions. For example, the reconstructed protein sequences 126 can correspond to single-domain antibodies having one or more CDRs and one or more framework regions. In at least some examples, the reconstructed protein sequences 126 can have a greater amount of diversity than the original amino acid sequences used to produce the sequence fragments that are assembled to generate the reconstructed protein sequences 126.

[00106] The reconstructed protein sequences 126 can be provided to the sequence analysis system 124. In one or more examples, the sequence analysis system 124 can include at least a portion of an autoencoder. The autoencoder can implement unsupervised machine learning techniques using an artificial neural network architecture. In various examples, the sequence analysis system 124 can include an encoding component 502. The encoding component 502 can include a number of computational layers. To illustrate, the encoding component 502 can include a number of convolutional layers with each computational layer comprising a number of nodes that each have at least one function and one or more weights. In one or more illustrative examples, at least a portion of the computational layers of the encoding component 502 can include fully connected layers. The encoding component 502 can produce code data that is a representation of input data provided to the encoding component 502. For example, the encoding component 502 can produce code data that is a compressed version of the reconstructed protein sequences 126.

[00107] In one or more examples, the encoding component 502 can include a trained encoder of an autoencoder that has undergone a training process using training data that corresponds to amino acid sequences related to the reconstructed protein sequences 126. For example, in scenarios where the reconstructed protein sequences 126 correspond to amino acid sequences of single-domain antibodies, the amino acid sequences included in the training data for the encoding component 502 can also correspond to amino acid sequences of single-domain antibodies. In at least some examples, the training data used to train the encoding component 502 can be similar to or the same as the training data used to train the trained generative machine learning component 110. The encoding component 502 can include at least one of encoder computer-readable instructions, encoder logic, or encoder circuitry.

[00108] In the illustrative example of Figure 5, the encoding component 502 can produce code data that is representative of one or more reconstructed protein sequences 126. The code data generated by the encoding component 502 based on a reconstructed protein sequence 126 can comprise a sequence seed 504 that is provided to the generative machine learning architecture 102. The trained generative machine learning component 110 of the generative machine learning architecture 102 can produce a generated sequence 506 based on the sequence seed 504. In one or more examples, the trained generative machine learning component 110 can include a generating component of a generative adversarial network.

[00109] The generated sequence 506 can be provided to the sequence analysis system 124. The sequence analysis system 124 can analyze the generated sequence 506 with respect to one or more additional amino acid sequences. In one or more examples, the sequence analysis system 124 can analyze the generated sequence 506 with respect to one or more reconstructed protein sequences 126. In one or more illustrative examples, the sequence analysis system 124 can analyze the generated sequence 506 with respect to the reconstructed protein sequence 126 used to generate the sequence seed 504. In one or more additional examples, the sequence analysis system 124 can analyze the generated sequence 506 in relation to additional amino acid sequences generated by the trained generative machine learning component 110. In one or more further examples, the sequence analysis system 124 can analyze the generated sequence 506 with respect to amino acid sequences used to train the trained generative machine learning component 110.

[00110] In various examples, the sequence analysis system 124 can analyze the generated sequence 506 with respect to additional amino acid sequences to determine an amount of homology between one or more regions of the generated sequence 506 and one or more regions of the additional amino acid sequences. In addition, the sequence analysis system 124 can analyze the generated sequence 506 to determine whether the generated sequence 506 has one or more characteristics of the additional amino acid sequences. For example, the sequence analysis system 124 can analyze the generated sequence 506 to determine whether the generated sequence 506 includes one or more CDRs or one or more framework regions. In one or more illustrative examples, the sequence analysis system 124 can analyze the generated sequence 506 to determine an amount of homology between one or more regions of the generated sequence 506 and at least one of one or more CDRs or one or more framework regions of the additional amino acid sequences.

[00111] In at least some examples, the sequence analysis system 124 can analyze the generated sequence 506 to determine a measure of similarity between the generated sequence 506 and amino acid sequences generated by the trained generative machine learning component 110. In situations where the measure of similarity of the generated sequence 506 is less than a threshold level, the generated sequence 506 can be excluded from a library of amino acid sequences that are candidates for being synthesized and used as a treatment for a biological condition. In this way, the sequence analysis system 124 operates to minimize the possibility of the generated sequence 506 being added to a library of sequences when the generated sequence 506 does not have at least one of one or more expected features or one or more expected characteristics of proteins that correspond to amino acid sequences included in the library. To illustrate, the reconstructed protein sequences 126 and a library of amino acid sequences can correspond to human and/or humanized single-domain antibodies. In these scenarios, the sequence analysis system 124 can analyze the generated sequence 506 to determine a likelihood that the generated sequence 506 corresponding to a human and/or humanized single-domain antibody.

[00112] Figures 6 and 7 illustrate example processes for generating amino acid sequences of proteins using machine learning techniques. The example processes are illustrated as collections of blocks in logical flow graphs, which represent sequences of operations that can be implemented in hardware, software, or a combination thereof. The blocks are referenced by numbers. In the context of software, the blocks represent computer-executable instructions stored on one or more computer-readable media that, when executed by one or more processing units (such as hardware microprocessors), perform the recited operations. Generally, computer-executable instructions include routines, programs, objects, components, data structures, and the like that perform particular functions or implement particular data types. The order in which the operations are described is not intended to be construed as a limitation, and any number of the described blocks can be combined in any order and/or in parallel to implement the process.

[00113] Figure 6 is a flow diagram illustrating an example process 600 to assemble amino acid sequences of using fragments of additional amino acid sequences previously generated using a generative adversarial network, in accordance with one or more some implementations. The process 600 can include, at 602, generating, using a generative adversarial network, a plurality of amino acid sequences that correspond to at least one of antibodies or antibody segments. In one or more examples, the plurality of amino acid sequences can correspond to amino acid sequences of single-domain antibodies. In one or more illustrative examples, the amino acid sequences can correspond to amino acid sequences of VHHs.

[00114] In addition, the process 600 can include, at 604, producing a plurality of fragments of the individual amino acids by dividing individual amino acid sequences into a number of sections. Individual fragments of the plurality of fragments can include one or more CDRs. In addition, individual fragments of the plurality of fragments can include at least a portion of one or more framework regions. In various examples, at least a portion of the framework regions bordering a CDR included in an individual fragment can be conserved. In one or more examples, at least a portion of the amino acid sequences of individual fragments can include overlapping positions with respect to amino acid sequences of other individual fragments.

[00115] Further, at 606, the process 600 can include assembling the plurality of fragments to generate reconstructed amino acid sequences. The plurality of fragments can be assembled by combining fragments that correspond to different initial amino acid sequences to generate new amino acid sequences that are different from the original plurality of amino acid sequences generated by the generative adversarial network. In one or more examples, the reconstructed amino acid sequences can be assembled according to at least one of one or more rules, one or more schema, or one or more frameworks. In one or more illustrative examples, the reconstructed sequences can include at least a first CDR, a second CDR, and a third CDR.

In various examples, the first CDR, the second CDR, and the third CDR can correspond to CDRs of camelid single-domain antibodies. The reconstructed sequences can also include at least a first framework region, a second framework region, a third framework region, and a fourth framework region.

[00116] In at least some examples, the plurality of amino acid sequences generated by the generative adversarial network can correspond to humanized amino acid sequences derived from amino acid sequences of antibodies that are produced by non-human mammals. In one or more additional examples, the plurality of amino acid sequences generated by the generative adversarial network can correspond to amino acid sequences of antibodies produced by non-human mammals. In these scenarios, the reconstructed amino acid sequences can be humanized. In various examples, the humanization process can be performed with regard to amino acid sequences of germline human antibodies, such that a number of positions of the amino acid sequences of the non-human antibodies are modified to correspond to the amino acids at corresponding positions of the amino acid sequences of the germline human antibodies.

[00117] In various examples, the generative adversarial network can include multiple generating components. For example, the generative adversarial network can be trained using training amino acid sequences derived from non-humanized single-domain antibodies produced by one or more non-human mammals. In these instances, the training amino acid sequences can be divided into sub-groups that include conserved sequences in framework regions between CDRs of the training amino acid sequences. To illustrate, the training amino acid sequences can include three CDRs and can be divided into a first sub-grouping that includes the first CDR and the second CDR with one or more conserved amino acids in the framework region between the first CDR and the second CDR and a second sub-grouping that includes the second CDR and the third CDR with one or more conserved amino acids in an additional framework region between the second CDR and the third CDR. In these scenarios, the first sub-grouping can comprise amino acid sequences that can be used to train a first generating component of the generative adversarial network and the second sub-grouping can comprise amino acid sequences that can be used to train a second generating component of the generative adversarial network. After the training process for the first generating component and the second generative component is complete, first amino acid sequences produced by the first generating component can be combined with second amino acid sequences produced by the second generating component. The combined amino acid sequences can then be humanized according to one or more template amino acid sequences that correspond to amino acid sequences of at least one of antibodies or antibody segments produced in accordance with human germline genomic regions.

[00118] In one or more additional examples, the reconstructed amino acid sequences can be analyzed with respect to the plurality of amino acid sequences generated by the generative adversarial network. In one or more examples, an autoencoder can generate seed data using a reconstructed amino acid sequence and feed the seed data back into the generative adversarial network to generate an additional amino acid sequence. In situations where the reconstructed amino acid sequence and the additional amino acid sequence have a threshold amount of similarity, the reconstructed amino acid sequence can be a candidate for an antibody sequence library. In scenarios where the reconstructed amino acid sequence and the additional amino acid sequence have less than the threshold amount of similarity, the reconstructed amino acid sequence may be excluded from an antibody sequence library.

[00119] Figure 7 is a flow diagram illustrating an example process 700 to generate reconstructed amino acid sequences using fragments of humanized antibody sequences produced using a generative machine learning architecture, in accordance with one or more implementations. At 702, the process 700 can include obtaining amino acid sequences of segments of antibodies that are produced by one or more non-human mammals and that correspond to an antibody functional region. In one or more examples, the antibody functional region can include at least one CDR. Additionally, the antibodies can be produced by one or more camelids.

[00120] The process 700 can also include, at 704, modifying the amino acid sequences based on template amino acid sequences to generate humanized amino acid sequence. The template amino acid sequences can include antibody sequences that correspond to human germline genomic regions that most closely correspond to the amino acid sequences. In at least some examples, the template amino acid sequences can be determining by analyzing the amino acid sequences with respect to the template amino acid sequences to determine a measure of similarity between the amino acid sequences and the template amino acid sequences. A template amino acid sequence having a highest measure of similarity with respect to at least a portion of the amino acid sequences can be determined to generate the humanized amino acid sequences. In various examples, the humanized amino acid sequences can be generated by modifying one or more amino acids at one or more positions of the amino acid sequences based on one or more different amino acids at the one or more positions of the template amino acid sequences. In one or more examples, at least a portion of the amino acid sequences can be conserved when generating the humanized amino acid sequences. For example, amino acids included in CDRs of the amino acid sequences can be conserved when producing the humanized amino acid sequences.

[00121] At 706, the process 700 can include generating training data that includes the humanized amino acid sequences. In addition, at 708, the process 700 can include performing, using the training data, a training process for a generative machine learning architecture to produce a machine learning model that generates amino acid sequences that correspond to humanized antibody fragments. Further, at 710, the process 700 can include generating, using the machine learning model, a plurality of amino acid sequences. In at least some examples, the plurality of amino acid sequences can be referred to as GAN-generated amino acid sequences.

[00122] The process 700 can include, at 712, producing a plurality of fragments for individual amino acid sequences of the plurality of amino acid sequences. In one or more examples, the plurality of amino acid sequences can be divided into a number of sections. In at least some examples, the number of sections can be produced according to at least one of one or more rules, one or more schemas, or one or more frameworks. For example, individual amino acid sequences of the plurality of amino acid sequences can be divided into a plurality of sections such that each section includes at least one CDR and at least a portion of one or more framework regions of the individual amino acid sequences. In one or more illustrative examples, individual amino acid sequences can be divided into three sections with each section including a CDR and at least a portion of a framework region. Individual fragments of the plurality of fragments can correspond to a continuous, sequential group of amino acids included in a section of a GAN-generated amino acid sequence.

[00123] At 714, the process 700 can include assembling the plurality of fragments to generate reconstructed amino acid sequences. In various examples, individual fragments originating from different initial amino acid sequences generated by the generative adversarial network can be combined to produce a reconstructed amino acid sequence. In one or more examples, a first fragment including a first CDR produced from a first initial GAN-generated amino acid sequence can be combined with a second fragment including a second CDR produced from a second initial GAN-generated amino acid sequence to generate a reconstructed amino acid sequence. The reconstructed amino acid sequences can also be evaluated with respect to the GAN-generated amino acid sequences to determine whether or not to include a reconstructed amino acid sequence as a candidate for an antibody sequence library. For example, reconstructed amino acid sequences that have at least a threshold measure of similarity with respect to the GAN-generated amino acid sequences can be candidates for inclusion in an antibody sequence library. In at least some examples, a physical antibody or physical antibody fragment that corresponds to a reconstructed amino acid sequence of the reconstructed amino acid sequences can be synthesized. In various examples, the synthesized antibody or physical antibody fragment can be provided as a treatment for a disease. In one or more illustrative examples, the disease can include a virus. In one or more additional examples, the disease can include cancer. In one or more further examples, the disease can include a bacterial infection.

[00124] Figure 8 illustrates a diagrammatic representation of a machine 800 in the form of a computer system within which a set of instructions may be executed for causing the machine 800 to perform any one or more of the methodologies discussed herein, according to an example, according to an example embodiment. Specifically, Figure 8 shows a diagrammatic representation of the machine 800 in the example form of a computer system, within which Instructions 802 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 800 to perform any one or more of the methodologies discussed herein may be executed. For example, the Instructions 802 may cause the machine 800 to implement the frameworks and architectures 100, 200, 300, 400, 500 described with respect to Figures 1, 2, 3, 4, and 5, respectively, and to execute the processes 600, 700 described with respect to Figures 6 and 7, respectively.

[00125] The Instructions 802 transform the general, non-programmed machine 800 into a particular machine 800 programmed to carry out the described and illustrated functions in the manner described. In alternative embodiments, the machine 800 operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine 800 may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 800 may comprise, but not be limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (PDA), an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the Instructions 802, sequentially or otherwise, that specify actions to be taken by the machine 800. Further, while only a single machine 800 is illustrated, the term "machine" shall also be taken to include a collection of machines 800 that individually or jointly execute the Instructions 802 to perform any one or more of the methodologies discussed herein.

[00126] Examples of computing device 800 can include logic, one or more components, circuits (e.g., modules), or mechanisms. Circuits are tangible entities configured to perform certain operations. In an example, circuits can be arranged (e.g., internally or with respect to external entities such as other circuits) in a specified manner. In an example, one or more computer systems (e.g., a standalone, client or server computer system) or one or more hardware processors (processors) can be configured by software (e.g., instructions, an application portion, or an application) as a circuit that operates to perform certain operations as described herein. In an example, the software can reside (1) on a non-transitory machine readable medium or (2) in a transmission signal. In an example, the software, when executed by the underlying hardware of the circuit, causes the circuit to perform the certain operations.

[00127] In an example, a circuit can be implemented mechanically or electronically. For example, a circuit can comprise dedicated circuitry or logic that is specifically configured to perform one or more techniques such as discussed above, such as including a special-purpose processor, a field programmable gate array (FPGA) or an application-specific integrated circuit (ASIC). In an example, a circuit can comprise programmable logic (e.g., circuitry, as encompassed within a general-purpose processor or other programmable processor) that can be temporarily configured (e.g., by software) to perform the certain operations. It will be appreciated that the decision to implement a circuit mechanically (e.g., in dedicated and permanently configured circuitry), or in temporarily configured circuitry (e.g., configured by software) can be driven by cost and time considerations.

[00128] Accordingly, the term "circuit" is understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily (e.g., transitorily) configured (e.g., programmed) to operate in a specified manner or to perform specified operations. In an example, given a plurality of temporarily configured circuits, each of the circuits need not be configured or instantiated at any one instance in time. For example, where the circuits comprise a general-purpose processor configured via software, the general-purpose processor can be configured as respective different circuits at different times. Software can accordingly configure a processor, for example, to constitute a particular circuit at one instance of time and to constitute a different circuit at a different instance of time.

[00129] In an example, circuits can provide information to, and receive information from, other circuits. In this example, the circuits can be regarded as being communicatively coupled to one or more other circuits. Where multiple of such circuits exist contemporaneously, communications can be achieved through signal transmission (e.g., over appropriate circuits and buses) that connect the circuits. In embodiments in which multiple circuits are configured or instantiated at different times, communications between such circuits can be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple circuits have access. For example, one circuit can perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further circuit can then, at a later time, access the memory device to retrieve and process the stored output. In an example, circuits can be configured to initiate or receive communications with input or output devices and can operate on a resource (e.g., a collection of information).

[00130] The various operations of method examples described herein can be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors can constitute processor-implemented circuits that operate to perform one or more operations or functions. In an example, the circuits referred to herein can comprise processor-implemented circuits.

[00131] Similarly, the methods described herein can be at least partially processor implemented. For example, at least some of the operations of a method can be performed by one or processors or processor-implemented circuits. The performance of certain of the operations can be distributed among the one or more processors, not only residing within a single machine, but deployed across a number of machines. In an example, the processor or processors can be located in a single location (e.g., within a home environment, an office environment or as a server farm), while in other examples the processors can be distributed across a number of locations.

[00132] The one or more processors can also operate to support performance of the relevant operations in a "cloud computing" environment or as a "software as a service."

[00133] (SaaS). For example, at least some of the operations can be performed by a group of computers (as examples of machines including processors), with these operations being accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., Application Program Interfaces (APIs).)

[00134] Example embodiments (e.g., apparatus, systems, or methods) can be implemented in digital electronic circuitry, in computer hardware, in firmware, in software, or in any combination thereof. Example embodiments can be implemented using a computer program product (e.g., a computer program, tangibly embodied in an information carrier or in a machine-readable medium, for execution by, or to control the operation of, data processing apparatus such as a programmable processor, a computer, or multiple computers).

[00135] A computer program can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a software module, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

[00136] In an example, operations can be performed by one or more programmable processors executing a computer program to perform functions by operating on input data and generating output. Examples of method operations can also be performed by, and example apparatus can be implemented as, special purpose logic circuitry (e.g., a field programmable gate array (FPGA) or an application-specific integrated circuit (ASIC)).

[00137] The computing system can include clients and servers. A client and server are generally remote from each other and generally interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In embodiments deploying a programmable computing system, it will be appreciated that both hardware and software architectures require consideration. Specifically, it will be appreciated that the choice of whether to implement certain functionality in permanently configured hardware (e.g., an ASIC), in temporarily configured hardware (e.g., a combination of software and a programmable processor), or a combination of permanently and temporarily configured hardware can be a design choice. Below are set out hardware (e.g., computing device 800) and software architectures that can be deployed in example embodiments.

[00138] In an example, the computing device 800 can operate as a standalone device or the computing device 800 can be connected (e.g., networked) to other machines.

[00139] In a networked deployment, the computing device 800 can operate in the capacity of either a server or a client machine in server-client network environments. In an example, computing device 800 can act as a peer machine in peer-to-peer (or other distributed) network environments. The computing device 800 can be a personal computer (PC), a tablet PC, a set-top box (STB), a Personal Digital Assistant (PDA), a mobile telephone, a web appliance, a network router, switch or bridge, or any machine capable of executing instructions (sequential or otherwise) specifying actions to be taken (e.g., performed) by the computing device 800. Further, while only a single computing device 800 is illustrated, the term “computing device” shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[00140] Example computing device 800 can include a processor 804 (e.g., a central processing unit CPU), a graphics processing unit (GPU) or both), a main memory 806 and a static memory 808, some or all of which can communicate with each other via a bus 810. The computing device 800 can further include a display unit 812, an alphanumeric input device 814 (e.g., a keyboard), and a user interface (UI) navigation device 816 (e.g., a mouse). In an example, the display unit 812, input device 814 and UI navigation device 816 can be a touch screen display. The computing device 800 can additionally include a storage device (e.g., drive unit) 818, a signal generation device 820 (e.g., a speaker), a network interface device 822, and one or more sensors 824, such as a global positioning system (GPS) sensor, compass, accelerometer, or another sensor.

[00141] The storage device 818 can include a machine readable medium 826 on which is stored one or more sets of data structures or Instructions 802 (e.g., software) embodying or utilized by any one or more of the methodologies or functions described herein. The Instructions 802 can also reside, completely or at least partially, within the main memory 806, within static memory 808, or within the processor 804 during execution thereof by the computing device 800. In an example, one or any combination of the processor 804, the main memory 806, the static memory 808, or the storage device 818 can constitute machine readable media.

[00142] While the machine readable medium 826 is illustrated as a single medium, the term "machine readable medium" can include a single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that configured to store the one or more Instructions 802. The term “machine readable medium” can also be taken to include any tangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of the present disclosure or that is capable of storing, encoding or carrying data structures utilized by or associated with such instructions. The term "machine readable medium" can accordingly be taken to include, but not be limited to, solid-state memories, and optical and magnetic media. Specific examples of machine-readable media can include non-volatile memory, including, by way of example, semiconductor memory devices (e.g., Electrically Programmable Read-Only Memory

[00143] (EPROM), Electrically Erasable Programmable Read-Only Memory (EEPROM)) and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks.

[00144] The instructions 802 can further be transmitted or received over a communications network 828 using a transmission medium via the network interface device 822 utilizing any one of a number of transfer protocols (e.g., frame relay, IP, TCP, UDP, HTTP, etc.). Example communication networks can include a local area network (LAN), a wide area network (WAN), a packet data network (e.g., the Internet), mobile telephone networks (e.g., cellular networks), Plain Old Telephone (POTS) networks, and wireless data networks (e.g., IEEE 802.11 standards family known as Wi-Fi®, IEEE 802.16 standards family known as WiMax®), peer-to-peer (P2P) networks, among others. The term "transmission medium" shall be taken to include any intangible medium that is capable of storing, encoding or carrying instructions for execution by the machine, and includes digital or analog communications signals or other intangible medium to facilitate communication of such software.

[00145] In view of the above-described implementations of subject matter this application discloses the following list of examples, wherein one feature of an example in isolation or more than one feature of an example, taken in combination and, optionally, in combination with one or more features of one or more further examples are further examples also falling within the disclosure of this application.

[00146] Example 1 is a method comprising: obtaining, by a computing system including one or more computing devices having one or more processors and memory, amino acid sequences of at least one of antibodies or antibody segments, the at least one of antibodies or antibody segments being produced by one or more non-human mammals, wherein at least a portion of the individual amino acid sequences correspond to an antibody functional region; modifying, by the computing system, the amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least a segment of an antibody produced by a human; generating, by the computing system, training data that includes the humanized amino acid sequences; performing, by the computing system and using the training data, a training process for a generative machine learning architecture to produce a machine learning model that generates amino acid sequences that correspond to humanized antibody fragments; generating, by the computing system and using the machine learning model, a plurality of amino acid sequences; producing, by the computing system and for individual amino acid sequences of the plurality of amino acid sequences, a plurality of fragments of the individual amino acid sequences; and assembling, by the computing system, the plurality of fragments to generate reconstructed amino acid sequences, at least a portion of the reconstructed amino acid sequences being different from the plurality of amino acid sequences.

[00147] In Example 2, the subject matter of example 1, includes: determining, by the computing system, a first portion of the plurality of fragments; producing, by the computing system, a first fragment pool that includes the first portion of the plurality of fragments; determining, by the computing system, a second portion of the plurality of fragments that is different from the first portion of the plurality of fragments; and producing, by the computing system, a second fragment pool that includes the second portion of the plurality of fragments.

[00148] In Example 3, the subject matter of example 2, includes the first portion of the plurality of fragments including a first complementarity determining region (CDR) and the second portion of the plurality of fragments including a second CDR.

[00149] In Example 4, the subject matter of example 3, includes the first portion of the plurality of fragments including at least a portion of a first framework region and the second portion of the plurality of fragments including at least a portion of a second framework region.

[00150] In Example 5, the subject matter of example 4, includes the first CDR corresponding to a first additional CDR of a camelid antibody and the second CDR corresponding to a second additional CDR of the camelid antibody.

[00151] In Example 6, the subject matter of example 5, includes the camelid antibody including a single-domain camelid antibody.

[00152] In Example 7, the subject matter of example 5, includes the first framework region including first amino acids that corresponds to a sequence of the camelid antibody framework region and second amino acids that correspond to a human germline framework region.

[00153] In Example 8, the subject matter of example 5, includes: the first portion of the plurality of fragments including the first framework region and a portion of the second framework region; the second portion of the plurality of fragments include the second framework region and a portion of a third framework region; and the portion of the second framework region included in the portion of the plurality of fragments overlaps with a subsection of the second framework region of the second portion of the plurality of fragments.

[00154] In Example 9, the subject matter of example 3, includes the reconstructed amino acid sequences being assembled using a first sequence from the first pool and a second sequence from the second pool, the first sequence and the second sequence of individual reconstructed amino acid sequences being different from the original plurality of amino acid sequences.

[00155] In Example 10, the subject matter of any one of examples 1-9, includes: determining, by the computing system, that an amino acid sequence includes a first cysteine amino acid at a first position of the amino acid sequence, the first cysteine amino acid being unpaired with another cysteine amino acid; and modifying, by the computing system, a non-cysteine amino acid located at a second position of the amino acid sequence to become a second cysteine amino acid such that the first cysteine amino acid is paired with the second amino acid in a humanized amino acid sequence.

[00156] In Example 11, the subject matter of any one of examples 1-10, includes the training process being a first training process and the humanized amino acid sequences corresponding to at least one of antibodies or antibody segments having a first set of structural features and a first set of biophysical properties, the first set of structural features including a structural feature corresponding to one or more first quantitative measures and the first set of biophysical properties including a biophysical property corresponding to one or more first additional quantitative measures.

[00157] In Example 12, the subject matter of example 11, includes: producing, by the computing system, additional training data including additional amino acid sequences that correspond to at least one of additional antibodies or additional antibody segments having a second set of structural features and a second set of biophysical properties, wherein: the structural feature corresponds to one or more second quantitative measures in the second set of structural features; the biophysical property corresponds to one or more second additional quantitative measures in the second set of biophysical properties; at least a portion of the second quantitative measures are different from at least a portion of the first quantitative measures; and at least a portion of the second additional quantitative measures are different from at least a portion of the first additional quantitative measures.

[00158] In Example 13, the subject matter of example 12, includes: performing, by the computing system, a second training process using the additional training data to generate a modified version of the machine learning model, wherein the modified version of the machine learning model generates further humanized amino acid sequences that correspond to at least one of antibodies or antibody segments having the one or more second quantitative measures for the structural property and the one or more second additional quantitative measures for the biophysical property; producing, by the computing system, a plurality of additional fragments for individual further amino acid sequences; and assembling, by the computing system, the plurality of additional fragments to generate additional reconstructed amino acid sequences.

[00159] Example 14 is a computing system comprising: one or more hardware processors; and one or more non-transitory computer readable media storing computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform operations comprising: obtaining amino acid sequences of at least one of antibodies or antibody segments, the at least one of antibodies or antibody segments being produced by one or more non-human mammals, wherein at least a portion of the individual amino acid sequences correspond to an antibody functional region; modifying the amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least a segment of an antibody produced by a human; generating training data that includes the humanized amino acid sequences; performing, using the training data, a training process for a generative machine learning architecture to produce a machine learning model that generates amino acid sequences that correspond to humanized antibody fragments; generating, using the machine learning model, a plurality of amino acid sequences; producing, for individual amino acid sequences of the plurality of amino acid sequences, a plurality of fragments of the individual amino acid sequences; and assembling, by the computing system, the plurality of fragments to generate reconstructed amino acid sequences, at least a portion of the reconstructed amino acid sequences being different from the plurality of amino acid sequences.

[00160] In Example 15, the subject matter of example 14, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: determining a first portion of the plurality of fragments; producing a first fragment pool that includes the first portion of the plurality of fragments; determining a second portion of the plurality of fragments that is different from the first portion of the plurality of fragments; and producing a second fragment pool that includes the second portion of the plurality of fragments.

[00161] In Example 16, the subject matter of example 15, includes the first portion of the plurality of fragments including a first complementarity determining region (CDR) and the second portion of the plurality of fragments include a second CDR.

[00162] In Example 17, the subject matter of example 16, includes the first portion of the plurality of fragments including at least a portion of a first framework region and the second portion of the plurality of fragments include at least a portion of a second framework region.

[00163] In Example 18, the subject matter of example 17, includes the first CDR corresponding to a first additional CDR of a camelid antibody and the second CDR corresponds to a second additional CDR of the camelid antibody.

[00164] In Example 19, the subject matter of example 18, includes the camelid antibody including a single-domain camelid antibody.

[00165] In Example 20, the subject matter of example 17, includes the first framework region including first amino acids that corresponds to a sequence of the camelid antibody framework region and second amino acids that correspond to a human germline framework region.

[00166] In Example 21, the subject matter of example 17, includes: the first portion of the plurality of fragments including the first framework region and a portion of the second framework region; the second portion of the plurality of fragments including the second framework region and a portion of a third framework region; and the portion of the second framework region included in the portion of the plurality of fragments overlaps with a subsection of the second framework region of the second portion of the plurality of fragments.

[00167] In Example 22, the subject matter of example 16, includes the reconstructed amino acid sequences being assembled using a first sequence from the first pool and a second sequence from the second pool, the first sequence and the second sequence of individual reconstructed amino acid sequences being different from the original plurality of amino acid sequences.

[00168] In Example 23, the subject matter of any one of examples 14-22, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: determining that an amino acid sequence includes a first cysteine amino acid at a first position of the amino acid sequence, the first cysteine amino acid being unpaired with another cysteine amino acid; and modifying a non-cysteine amino acid located at a second position of the amino acid sequence to become a second cysteine amino acid such that the first cysteine amino acid is paired with the second amino acid in a humanized amino acid sequence.

[00169] In Example 24, the subject matter of any one of examples 14-23, includes the training process being a first training process and the humanized amino acid sequences correspond to at least one of antibodies or antibody segments having a first set of structural features and a first set of biophysical properties, the first set of structural features including a structural feature corresponding to one or more first quantitative measures and the first set of biophysical properties including a biophysical property corresponding to one or more first additional quantitative measures.

[00170] In Example 25, the subject matter of example 24, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: producing additional training data including additional amino acid sequences that correspond to at least one of additional antibodies or additional antibody segments having a second set of structural features and a second set of biophysical properties, wherein: the structural feature corresponds to one or more second quantitative measures in the second set of structural features; the biophysical property corresponds to one or more second additional quantitative measures in the second set of biophysical properties; at least a portion of the second quantitative measures are different from at least a portion of the first quantitative measures; and at least a portion of the second additional quantitative measures are different from at least a portion of the first additional quantitative measures.

[00171] In Example 26, the subject matter of example 25, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: performing a second training process using the additional training data to generate a modified version of the machine learning model, wherein the modified version of the machine learning model generates further humanized amino acid sequences that correspond to at least one of antibodies or antibody segments having the one or more second quantitative measures for the structural property and the one or more second additional quantitative measures for the biophysical property; producing a plurality of additional fragments for individual further amino acid sequences; and assembling the plurality of additional fragments to generate additional reconstructed amino acid sequences.

[00172] Example 27 is a computing system comprising: one or more hardware processors; and one or more non-transitory computer readable media storing computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform operations comprising: generating, using one or more models of one or more generating components of a generative adversarial network, a plurality of amino acid sequences, the plurality of amino acid sequences corresponding to amino acid sequences of at least one of antibodies or antibody segments; producing a plurality of fragments of individual amino acid sequences of the plurality of amino acid sequences by dividing individual amino acid sequences into a number of sections; and assembling the plurality of fragments to generate reconstructed amino acid sequences, at least a portion of the reconstructed amino acid sequences being different from the plurality of amino acid sequences, individual reconstructed amino acid sequences corresponding to at least one of antibodies or antibody segments.

[00173] In Example 28. The computing system of claim 27, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: obtaining additional amino acid sequences of at least one of antibodies or antibody segments, the at least one of antibodies or antibody segments being produced by one or more non-human mammals, wherein at least a portion of the individual amino acid sequences correspond to an antibody functional region; and modifying the additional amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least a segment of an antibody produced by a human; generating training data that includes the humanized amino acid sequences; and performing, using the training data, a training process for the generative adversarial network to produce the one or more models of the one or more generating components of the generative adversarial network.

[00174] In Example 29, the subject matter of example 27, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: generating training data for the generative adversarial network, the training data including amino acid sequences of at least one of antibodies or antibody segments produced by one or more non-human mammals; and performing, using the training data, a training process for the generative adversarial network to produce the one or more models of the one or more generating components of the generative adversarial network.

[00175] In Example 30, the subject matter of example 29, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: modifying positions of the reconstructed amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least one of an antibody or antibody segment produced by a human.

[00176] In Example 31, the subject matter of any one of examples 27-30, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: providing a reconstructed amino acid sequence to an encoding component of an autoencoder; generating, using the encoding component, a sequence seed that corresponds to the reconstructed amino acid sequence; providing the sequence seed to the generative adversarial network; and generating, using the generative machine learning model and based on the sequence seed, a further amino acid sequence.

[00177] In Example 32, the subject matter of example 31, includes the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processor to perform additional operations comprising: analyzing the further amino acid sequence in relation training amino acid sequences used to train the generative adversarial network to determine a measure of similarity between the further amino acid sequence and the training amino acid sequences; and determining that the measure of similarity is at least a minimum measure of similarity, the minimum measure of similarity indicating a threshold for identifying candidate amino acid sequences to include in a library of amino acid sequences.

[00178] Example 33 is a method comprising: generating, by a computing system including one or more computing devices having one or more processors and memory and using one or more models of one or more generating components of a generative adversarial network, a plurality of amino acid sequences, the plurality of amino acid sequences corresponding to amino acid sequences of at least one of antibodies or antibody segments; producing, by the computing system, a plurality of fragments of individual amino acid sequences of the plurality of amino acid sequences by dividing individual amino acid sequences into a number of sections; and assembling, by the computing system, the plurality of fragments to generate reconstructed amino acid sequences, at least a portion of the reconstructed amino acid sequences being different from the plurality of amino acid sequences, individual reconstructed amino acid sequences corresponding to at least one of antibodies or antibody segments.

[00179] In Example 34, the subject matter of example 33, comprises: obtaining, by the computing system, additional amino acid sequences of at least one of antibodies or antibody segments, the at least one of antibodies or antibody segments being produced by one or more non-human mammals, wherein at least a portion of the individual amino acid sequences correspond to an antibody functional region; and modifying, by the computing system, the additional amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least a segment of an antibody produced by a human; generating, by the computing system, training data that includes the humanized amino acid sequences; and performing, by the computing system and using the training data, a training process for the generative adversarial network to produce the one or more models of the one or more generating components of the generative adversarial network.

[00180] In Example 35, the subject matter of example 33, comprises: generating, by the computing system, training data for the generative adversarial network, the training data including amino acid sequences of at least one of antibodies or antibody segments produced by one or more non-human mammals; and performing, by the computing system and using the training data, a training process for the generative adversarial network to produce the one or more models of the one or more generating components of the generative adversarial network.

[00181] In Example 36, the subject matter of example 35, comprises: modifying, by the computing system, positions of the reconstructed amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least one of an antibody or antibody segment produced by a human.

[00182] In Example 37, the subject matter of any one of examples 33-36, comprises: providing, by the computing system, a reconstructed amino acid sequence to an encoding component of an autoencoder; generating, by the computing system and using the encoding component, a sequence seed that corresponds to the reconstructed amino acid sequence; providing, by the computing system, the sequence seed to the generative adversarial network; and generating, by the computing system and using the generative machine learning model and based on the sequence seed, a further amino acid sequence.

[00183] In Example 38, the subject matter of example 37, comprises: analyzing, by the computing system, the further amino acid sequence in relation training amino acid sequences used to train the generative adversarial network to determine a measure of similarity between the further amino acid sequence and the training amino acid sequences; and determining, by the computing system, that the measure of similarity is at least a minimum measure of similarity, the minimum measure of similarity indicating a threshold for identifying candidate amino acid sequences to include in a library of amino acid sequences.

Examples

[00184] Antibody sequences and antibody fragment sequences were generated according to the techniques described herein. Additionally, antibodies and antibody fragments corresponding to the in-silico sequences were synthesized in a manner similar to that described herein, as well as the use of library construction techniques described in Loomis et al., AI-based antibody discovery platform identifies novel, diverse and pharmacologically active therapeutic antibodies against multiple SARS-CoV-2 strains. bioRxiv 2023.08.21.554197; doi: https://doi.org/10.1101/2023.08.21.554197.

[00185] Training of generative adversarial networks was performed using at least 400,000 antibody sequences per chain obtained from the Observed Antibody Space (OAS) database. Humanization was performed according to implementations described herein. Antibody sequences were synthesized using M13 bacteriophage and, in some cases, included 5 HV, 4 KV and 1 LV germlines. Humanized VHHs were displayed on S. cerevisiae and, in some cases, with 2 germlines, 2 disulfide and 2 FW variations.

[00186] The synthesized antibody sequence and antibody sequence fragments were evaluated according to a number of analytical tests. Biochemical characterization samples were prepared. Where applicable, differential scanning fluorimetry was performed according to UniProt Consortium. 2019. UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res 47:D506–D515. Where applicable, low pH stability was determined according to Roberts CJ, Das TK, Sahin E. 2011. Predicting solution aggregation rates for therapeutic proteins: approaches and challenges. Int J Pharm 418:318–333. Additionally, where applicable chemical unfolding was performed with some modifications according to Klein F, Diskin R, Scheid JF, Gaebler C, Mouquet H, Georgiev IS, Pancera M, Zhou T, Incesu RB, Fu BZ, Gnanapragasam PNP, Oliveira TY, Seaman MS, Kwong PD, Bjorkman PJ, Nussenzweig MC. 2013. Somatic mutations of the immunoglobulin framework are generally required for broad and potent HIV-1 neutralization. Cell 153:126–138. Further, where applicable, relative solubility was assessed according to UniProt Consortium. 2019. UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res 47:D506–D515. Where applicable, self-interaction nanoparticle spectroscopy (SINS) was performed according to Amaya M, Cheng H, Borisevich V, Navaratnarajah CK, Cattaneo R, Cooper L, Moore TW, Gaisina IN, Geisbert TW, Rong L, Broder CC. 2021. A recombinant Cedar virus based high-throughput screening assay for henipavirus antiviral discovery. Antiviral Res 193:105084.

[00187] Example results of binding affinity and efficacy data are shown in Figures 9 and 10. In Figure 9, an HcAb binding assessment is shown. Flow cytometry (FACS)-isolated yeast VHH were converted to three HcAb formats and clonally expressed in mammalian transient system. The three HcAb formats were VHH-Fc, Fc-G_4S-VHH, and Fc-$(G_4S)_2$-VHH. Location of VHH fusion with respect to the Fc impacted target in an AlphaLISA was determined. Suitable targets (dashed boxes) for an in vitro activity assay were identified. The active VHH candidates will be combined with IgG that are specific to another bacterial target to generate bispecifics.

[00188] In Figure 10, luciferase activity in cells was measured 24 hour post viral infection for two antibody sequences generated according to implementations described herein and synthesized according to techniques described herein. Both experimental antibodies were functional against the virus and one of the antibodies showed a therapeutic response similar to a clinical control antibody used to treat the viral infection. Example luciferase assay systems can include Promega; catalog number E1500.

CLAIMS

What is claimed is:

1. A method comprising:

obtaining, by a computing system including one or more computing devices having one or more processors and memory, amino acid sequences of at least one of antibodies or antibody segments, the at least one of antibodies or antibody segments being produced by one or more non-human mammals, wherein at least a portion of individual amino acid sequences correspond to an antibody functional region;

modifying, by the computing system, the amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least a segment of an antibody produced by a human;

generating, by the computing system, training data that includes the humanized amino acid sequences;

performing, by the computing system and using the training data, a training process for a generative machine learning architecture to produce a machine learning model that generates amino acid sequences that correspond to humanized antibody fragments;

generating, by the computing system and using the machine learning model, a plurality of amino acid sequences;

producing, by the computing system and for individual amino acid sequences of the plurality of amino acid sequences, a plurality of fragments of the individual amino acid sequences; and

assembling, by the computing system, the plurality of fragments to generate reconstructed amino acid sequences, at least a portion of the reconstructed amino acid sequences being different from the plurality of amino acid sequences.

2. The method of claim 1, comprising:

determining, by the computing system, a first portion of the plurality of fragments;

producing, by the computing system, a first fragment pool that includes the first portion of the plurality of fragments;

determining, by the computing system, a second portion of the plurality of fragments that is different from the first portion of the plurality of fragments; and

producing, by the computing system, a second fragment pool that includes the second portion of the plurality of fragments.

3. The method of claim 2, wherein the first portion of the plurality of fragments include a first complementarity determining region (CDR) and the second portion of the plurality of fragments include a second CDR.

4. The method of claim 3, wherein the first portion of the plurality of fragments include at least a portion of a first framework region and the second portion of the plurality of fragments include at least a portion of a second framework region.

5. The method of claim 4, wherein the first CDR corresponds to a first additional CDR of a camelid antibody and the second CDR corresponds to a second additional CDR of the camelid antibody.

6. The method of claim 5, wherein the camelid antibody includes a single-domain camelid antibody.

7. The method of claim 5, wherein the first framework region includes first amino acids that corresponds to a sequence of a camelid antibody framework region and second amino acids that correspond to a human germline framework region.

8. The method of claim 5, wherein:

the first portion of the plurality of fragments include the first framework region and a portion of the second framework region;

the second portion of the plurality of fragments include the second framework region and a portion of a third framework region; and

the portion of the second framework region included in the portion of the plurality of fragments overlaps with a subsection of the second framework region of the second portion of the plurality of fragments.

9. The method of claim 3, wherein the reconstructed amino acid sequences are assembled using a first sequence from a first pool and a second sequence from a second pool, the first sequence and the second sequence comprising individual reconstructed amino acid sequences that are different from the amino acid sequences.

10. The method of claim 1, comprising:

determining, by the computing system, that an amino acid sequence includes a first cysteine amino acid at a first position of the amino acid sequence, the first cysteine amino acid being unpaired with another cysteine amino acid; and

modifying, by the computing system, a non-cysteine amino acid located at a second position of the amino acid sequence to become a second cysteine amino acid such that the first cysteine amino acid is paired with the second cysteine amino acid in a humanized amino acid sequence.

11. The method of claim 1, wherein the training process is a first training process and the humanized amino acid sequences correspond to at least one of antibodies or antibody segments having a first set of structural features and a first set of biophysical properties, the first set of structural features including a structural feature corresponding to one or more first quantitative measures and the first set of biophysical properties including a biophysical property corresponding to one or more first additional quantitative measures.

12. The method of claim 11, comprising:

producing, by the computing system, additional training data including additional amino acid sequences that correspond to at least one of additional antibodies or additional antibody segments having a second set of structural features and a second set of biophysical properties, wherein:

the structural feature corresponds to one or more second quantitative measures in the second set of structural features;

the biophysical property corresponds to one or more second additional quantitative measures in the second set of biophysical properties;

at least a portion of the one or more second quantitative measures are different from at least a portion of the one or more first quantitative measures; and at least a portion of the one or more second additional quantitative measures are different from at least a portion of the one or more first additional quantitative measures.

13. The method of claim 12, comprising:

performing, by the computing system, a second training process using the additional training data to generate a modified version of the machine learning model, wherein the modified version of the machine learning model generates further humanized amino acid sequences that correspond to at least one of antibodies or antibody segments having the one or more second quantitative measures for the structural feature and the one or more second additional quantitative measures for the biophysical property;

producing, by the computing system, a plurality of additional fragments for individual further amino acid sequences; and

assembling, by the computing system, the plurality of additional fragments to generate additional reconstructed amino acid sequences.

14. The method of claim 1, comprising:

synthesizing a physical antibody or physical antibody fragment that corresponds to a reconstructed amino acid sequence of the reconstructed amino acid sequences.

15. The method of claim 14, comprising:

providing the physical antibody or the physical antibody fragment as a treatment for a disease.

16. A computing system comprising:

one or more hardware processors; and

one or more non-transitory computer readable media storing computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more processors to perform operations comprising:

generating, using one or more models of one or more generating components of a generative adversarial network, a plurality of amino acid sequences, the plurality of amino acid sequences corresponding to amino acid sequences of at least one of antibodies or antibody segments;

producing a plurality of fragments of individual amino acid sequences of the plurality of amino acid sequences by dividing individual amino acid sequences into a number of sections; and

assembling the plurality of fragments to generate reconstructed amino acid sequences, at least a portion of the reconstructed amino acid sequences being different from the plurality of amino acid sequences, individual reconstructed amino acid sequences corresponding to at least one of antibodies or antibody segments.

17. The computing system of claim 16, wherein the one or more non-transitory computer readable media store additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more hardware processors to perform additional operations comprising:

obtaining additional amino acid sequences of at least one of antibodies or antibody segments, the at least one of antibodies or antibody segments being produced by one or more non-human mammals, wherein at least a portion of the individual amino acid sequences correspond to an antibody functional region; and

modifying the additional amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least a segment of an antibody produced by a human;

generating training data that includes the humanized amino acid sequences; and

performing, using the training data, a training process for the generative adversarial network to produce the one or more models of the one or more generating components of the generative adversarial network.

18. The computing system of claim 16, wherein the one or more non-transitory computer readable media store additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more hardware processors to perform additional operations comprising:

generating training data for the generative adversarial network, the training data including amino acid sequences of at least one of antibodies or antibody segments produced by one or more non-human mammals; and performing, using the training data, a training process for the generative adversarial network to produce the one or more models of the one or more generating components of the generative adversarial network.

19. The computing system of claim 18, wherein the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more hardware processors to perform additional operations comprising:

modifying positions of the reconstructed amino acid sequences based on one or more template amino acid sequences to generate humanized amino acid sequences, individual template amino acid sequences of the one or more template amino acid sequences corresponding to a germline sequence corresponding to at least one of an antibody or antibody segment produced by a human.

20. The computing system of claim 16, wherein the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more hardware processors to perform additional operations comprising:

providing a reconstructed amino acid sequence to an encoding component of an autoencoder;

generating, using the encoding component, a sequence seed that corresponds to the reconstructed amino acid sequence;

providing the sequence seed to the generative adversarial network; and

generating, using the generative adversarial network and based on the sequence seed, a further amino acid sequence.

21. The system of claim 20, wherein the one or more non-transitory computer readable media storing additional computer-executable instructions that, when executed by the one or more hardware processors, cause the one or more hardware processors to perform additional operations comprising:

analyzing the further amino acid sequence in relation training amino acid sequences used to train the generative adversarial network to determine a measure of similarity between the further amino acid sequence and the training amino acid sequences; and determining that the measure of similarity is at least a minimum measure of similarity, the minimum measure of similarity indicating a threshold for identifying candidate amino acid sequences to include in a library of amino acid sequences.

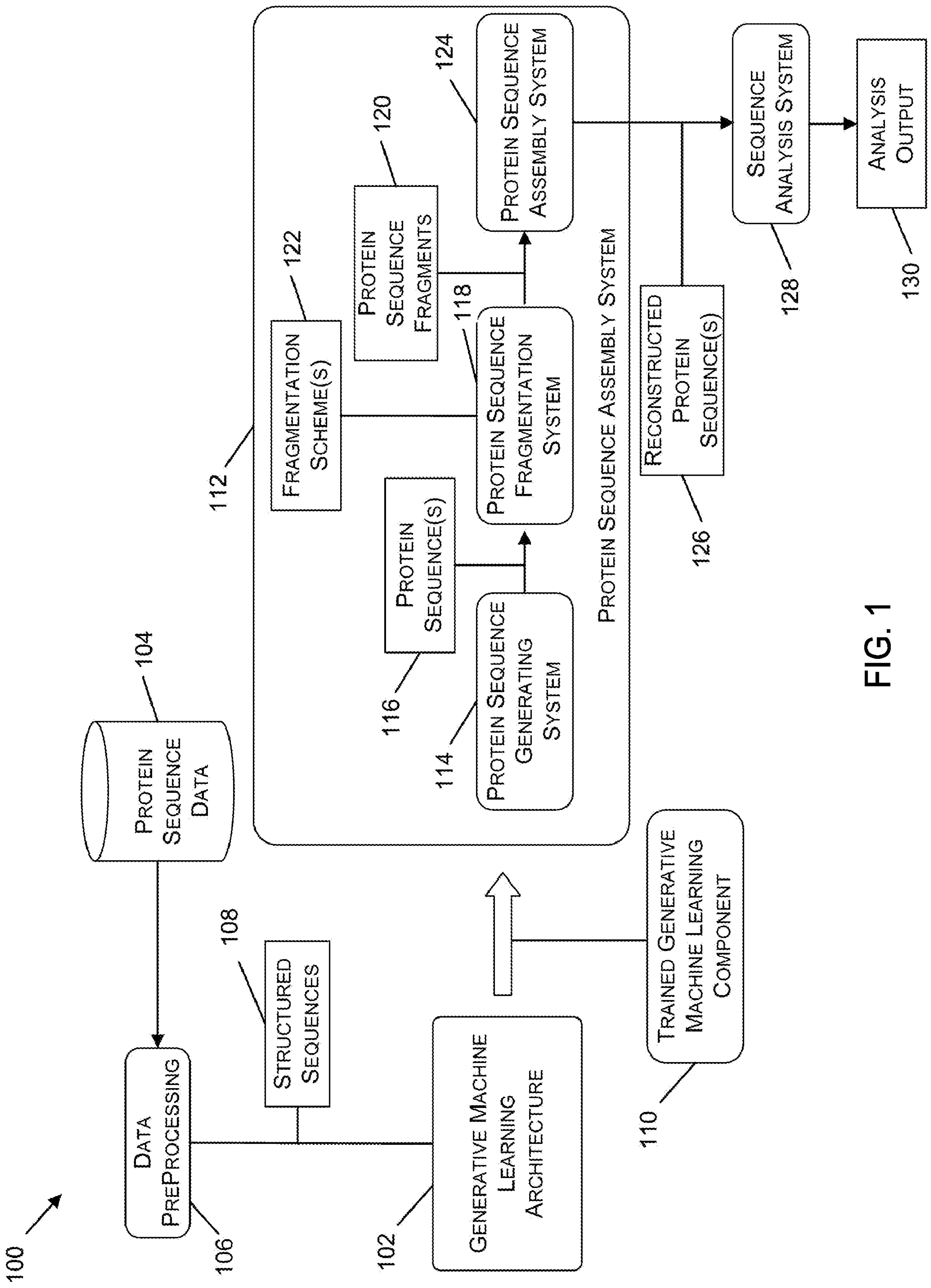

100
Data PreProcessing
106
Protein Sequence Data
104
Structured Sequences
108
Generative Machine Learning Architecture
102
Trained Generative Machine Learning Component
110
Protein Sequence Assembly System
112
Protein Sequence Generating System
114
Protein Sequence(s)
116
Protein Sequence Fragmentation System
118
Fragmentation Scheme(s)
122
Protein Sequence Fragments
120
Protein Sequence Assembly System
124
Reconstructed Protein Sequence(s)
126
Sequence Analysis System
128
Analysis Output
130

FIG. 1

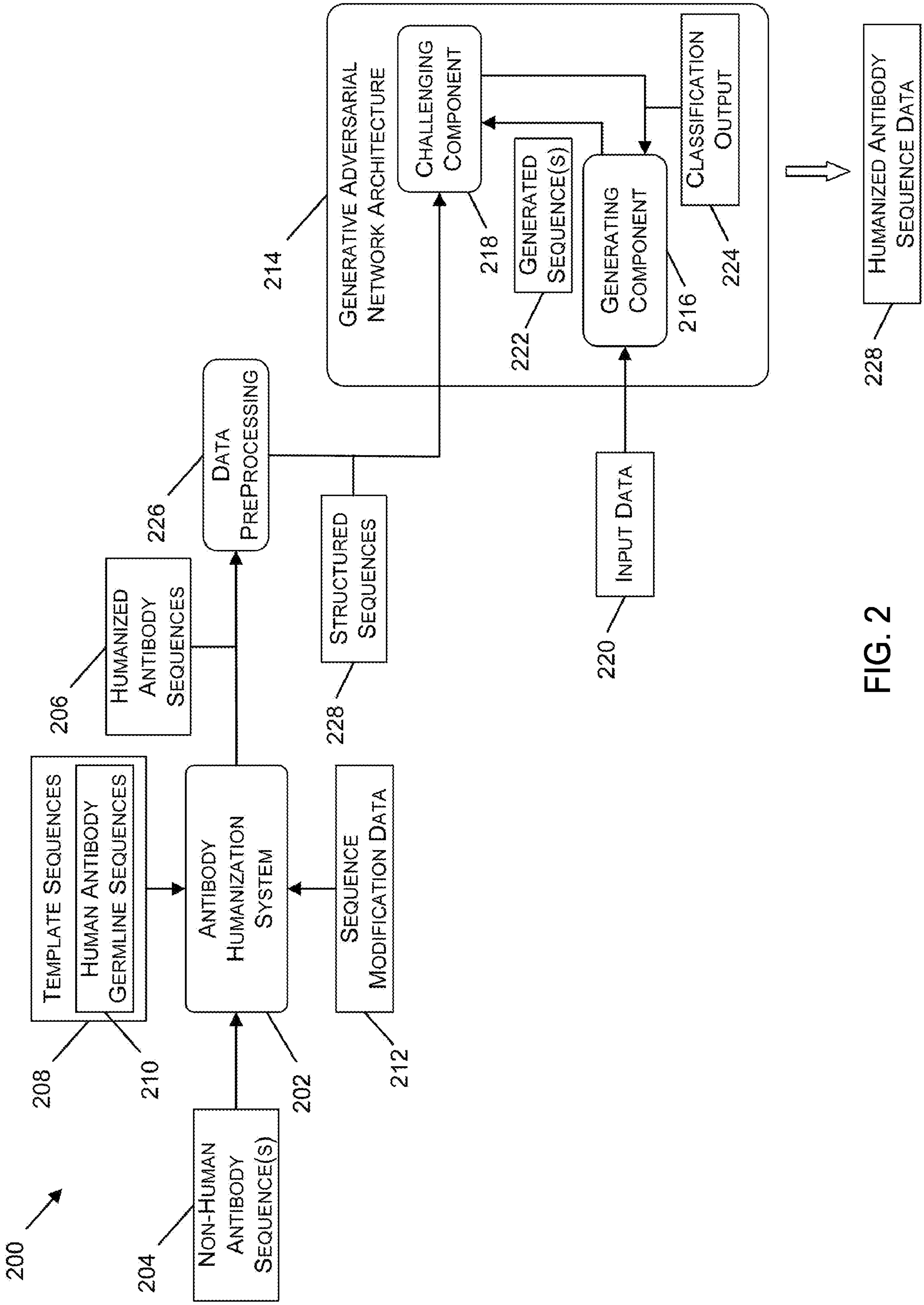

200
Non-Human Antibody Sequence(s)
204
Template Sequences
Human Antibody Germline Sequences
208
210
Antibody Humanization System
202
Sequence Modification Data
212
Humanized Antibody Sequences
206
Data PreProcessing
226
Structured Sequences
228
Generative Adversarial Network Architecture
214
Challenging Component
218
Generated Sequence(s)
222
Generating Component
216
Classification Output
224
Input Data
220
Humanized Antibody Sequence Data
228

FIG. 2

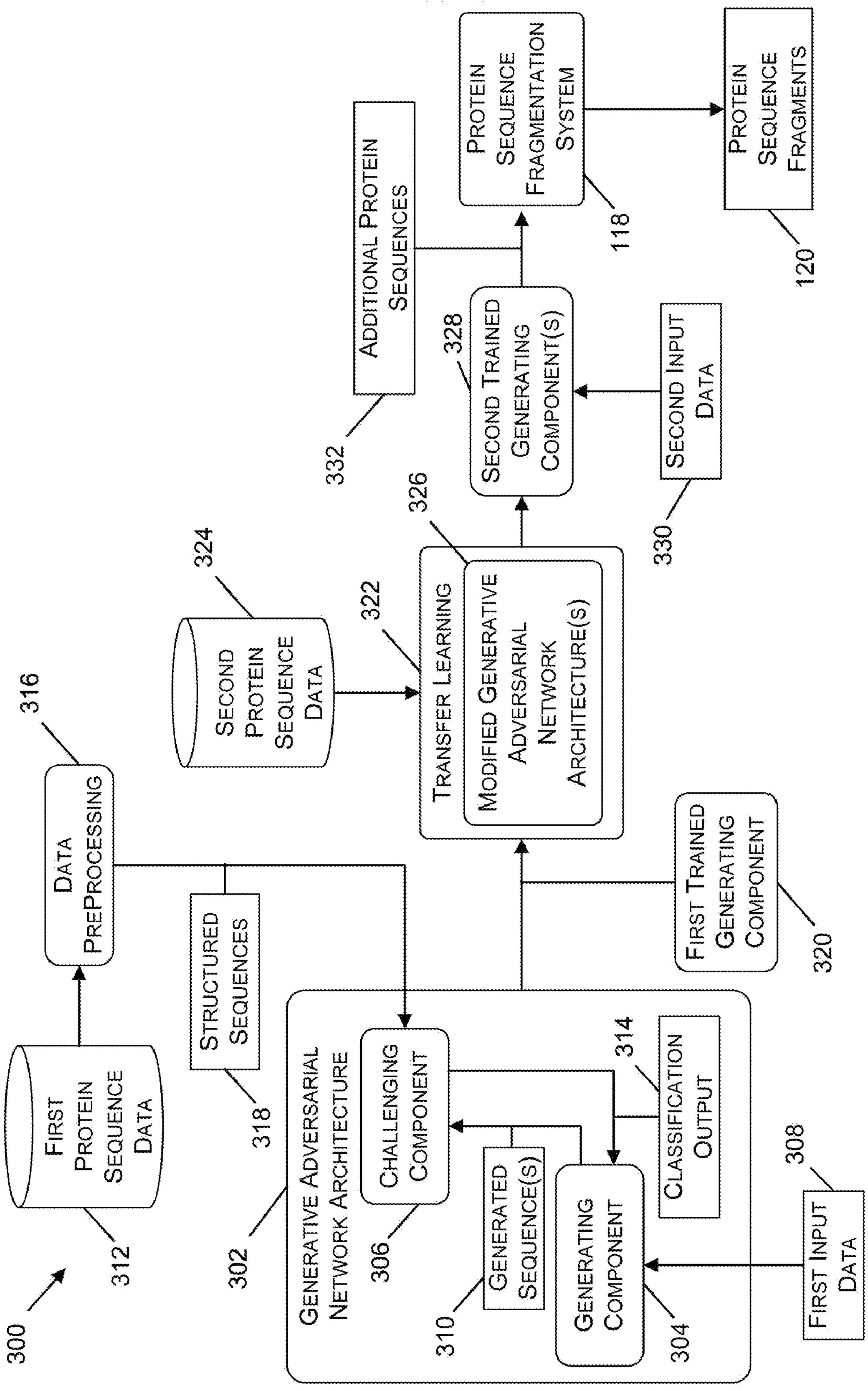

300
First Protein Sequence Data
312
Data PreProcessing
316
Structured Sequences
318
Generative Adversarial Network Architecture
302
Challenging Component
306
Generated Sequence(s)
310
Generating Component
304
Classification Output
314
First Input Data
308
First Trained Generating Component
320
Second Protein Sequence Data
324
Transfer Learning
322
Modified Generative Adversarial Network Architecture(s)
326
Second Trained Generating Component(s)
328
Second Input Data
330
Additional Protein Sequences
332
Protein Sequence Fragmentation System
118
Protein Sequence Fragments
120

FIG. 3

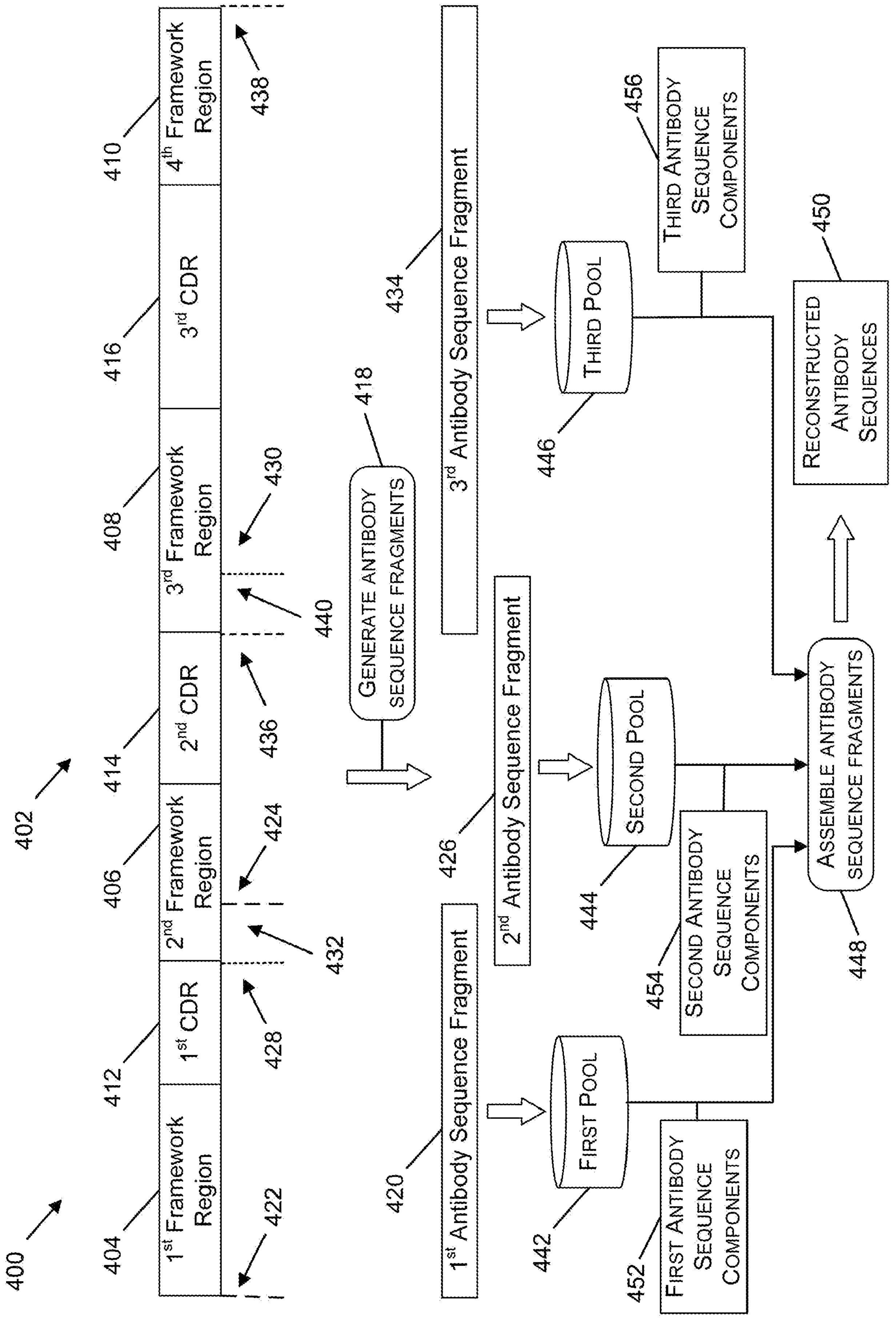

400
402
404
1st Framework Region
412
1st CDR
406
2nd Framework Region
414
2nd CDR
408
3rd Framework Region
416
3rd CDR
410
4th Framework Region
422
428
432
424
436
440
430
438
418
Generate Antibody Sequence Fragments
420
1st Antibody Sequence Fragment
426
2nd Antibody Sequence Fragment
434
3rd Antibody Sequence Fragment
442
First Pool
444
Second Pool
446
Third Pool
452
First Antibody Sequence Components
454
Second Antibody Sequence Components
456
Third Antibody Sequence Components
448
Assemble Antibody Sequence Fragments
450
Reconstructed Antibody Sequences

FIG. 4

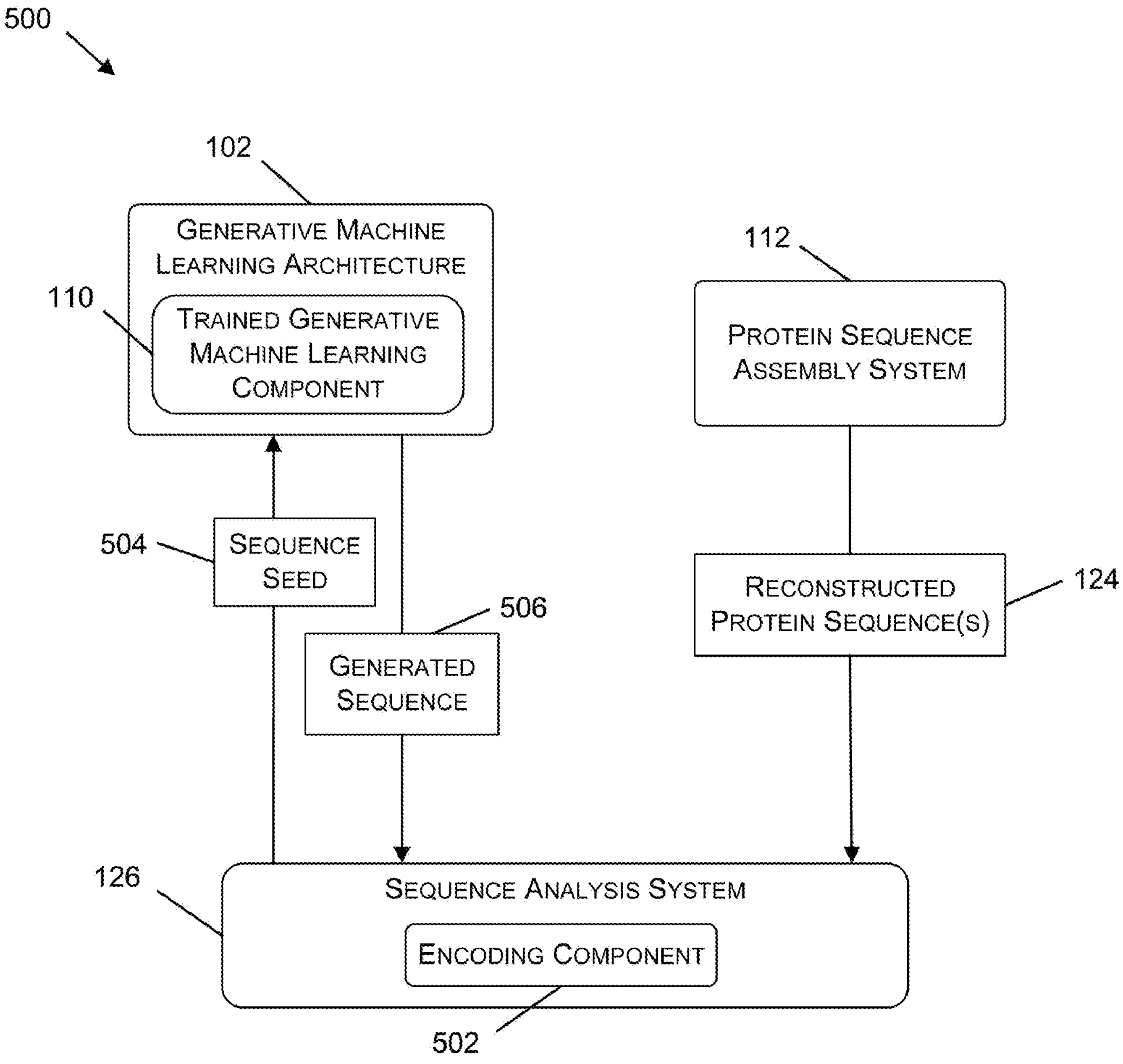

500
102
Generative Machine Learning Architecture
110
Trained Generative Machine Learning Component
112
Protein Sequence Assembly System
504
Sequence Seed
506
Generated Sequence
Reconstructed Protein Sequence(s)
124
126
Sequence Analysis System
Encoding Component
502

FIG. 5

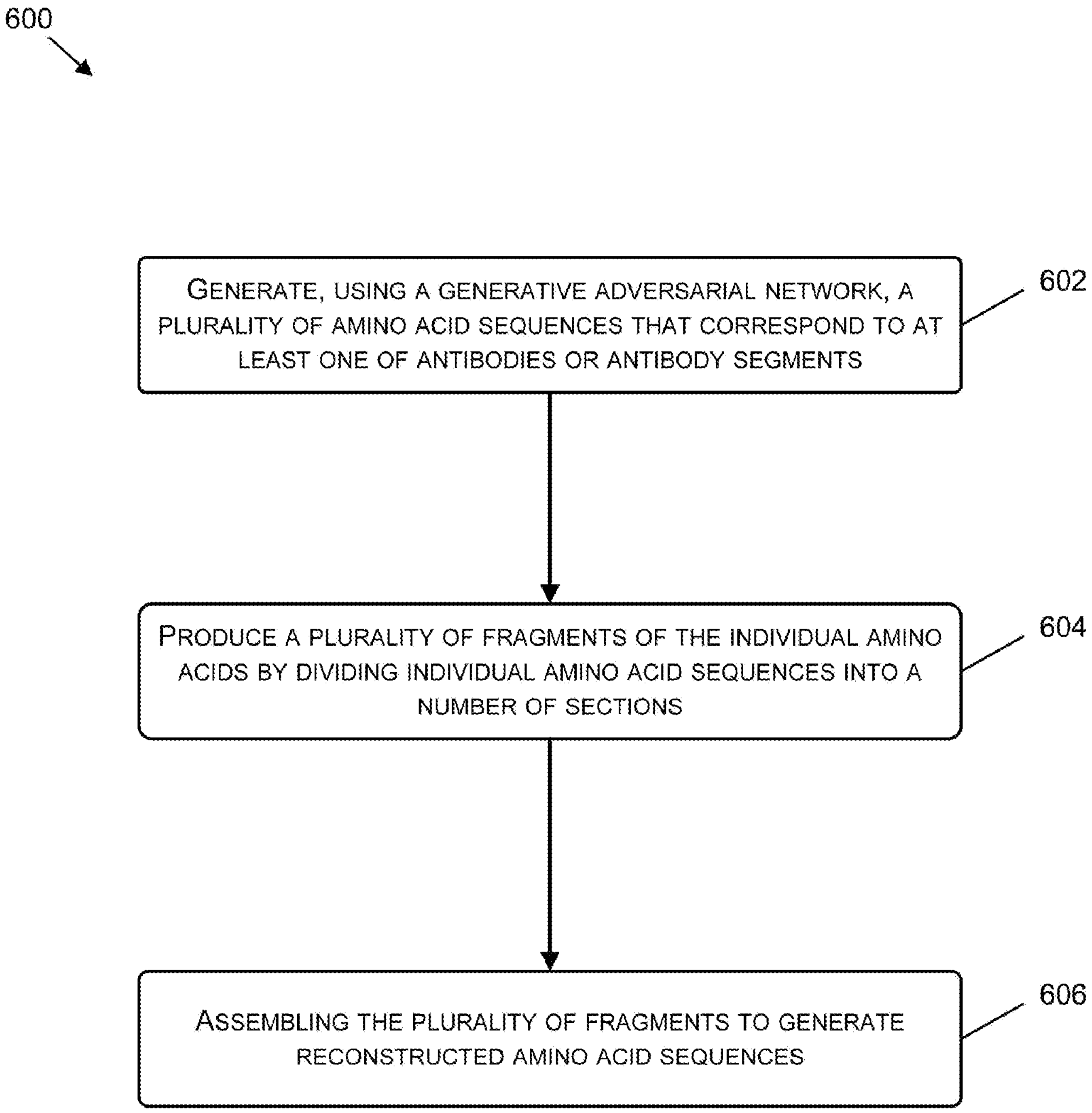

600
GENERATE, USING A GENERATIVE ADVERSARIAL NETWORK, A PLURALITY OF AMINO ACID SEQUENCES THAT CORRESPOND TO AT LEAST ONE OF ANTIBODIES OR ANTIBODY SEGMENTS
602
PRODUCE A PLURALITY OF FRAGMENTS OF THE INDIVIDUAL AMINO ACIDS BY DIVIDING INDIVIDUAL AMINO ACID SEQUENCES INTO A NUMBER OF SECTIONS
604
ASSEMBLING THE PLURALITY OF FRAGMENTS TO GENERATE RECONSTRUCTED AMINO ACID SEQUENCES
606

FIG. 6

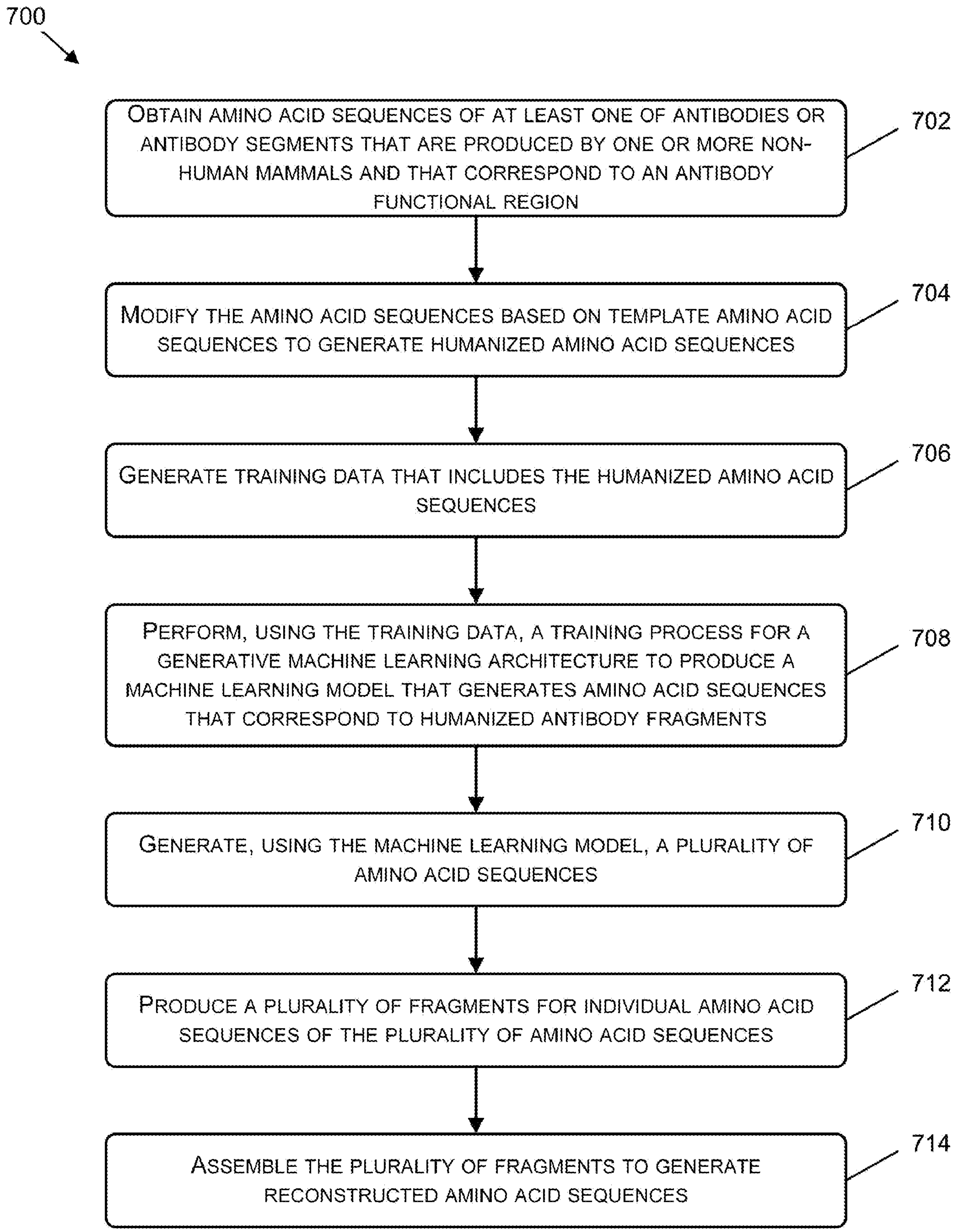

700
OBTAIN AMINO ACID SEQUENCES OF AT LEAST ONE OF ANTIBODIES OR ANTIBODY SEGMENTS THAT ARE PRODUCED BY ONE OR MORE NON-HUMAN MAMMALS AND THAT CORRESPOND TO AN ANTIBODY FUNCTIONAL REGION
702
MODIFY THE AMINO ACID SEQUENCES BASED ON TEMPLATE AMINO ACID SEQUENCES TO GENERATE HUMANIZED AMINO ACID SEQUENCES
704
GENERATE TRAINING DATA THAT INCLUDES THE HUMANIZED AMINO ACID SEQUENCES
706
PERFORM, USING THE TRAINING DATA, A TRAINING PROCESS FOR A GENERATIVE MACHINE LEARNING ARCHITECTURE TO PRODUCE A MACHINE LEARNING MODEL THAT GENERATES AMINO ACID SEQUENCES THAT CORRESPOND TO HUMANIZED ANTIBODY FRAGMENTS
708
GENERATE, USING THE MACHINE LEARNING MODEL, A PLURALITY OF AMINO ACID SEQUENCES
710
PRODUCE A PLURALITY OF FRAGMENTS FOR INDIVIDUAL AMINO ACID SEQUENCES OF THE PLURALITY OF AMINO ACID SEQUENCES
712
ASSEMBLE THE PLURALITY OF FRAGMENTS TO GENERATE RECONSTRUCTED AMINO ACID SEQUENCES
714

FIG. 7

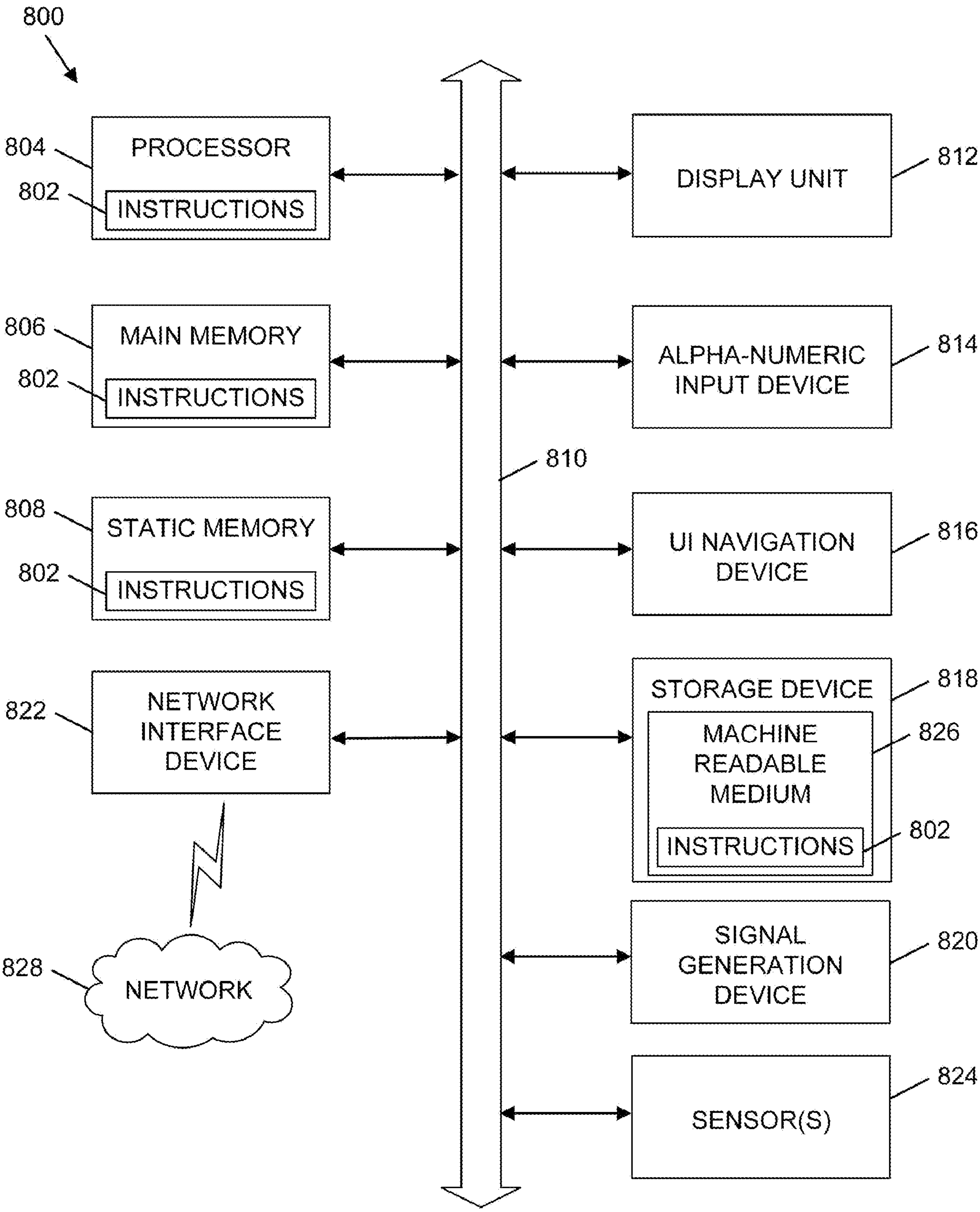

800
804
PROCESSOR
802
INSTRUCTIONS
806
MAIN MEMORY
802
INSTRUCTIONS
808
STATIC MEMORY
802
INSTRUCTIONS
822
NETWORK INTERFACE DEVICE
828
NETWORK
810
DISPLAY UNIT
812
ALPHA-NUMERIC INPUT DEVICE
814
UI NAVIGATION DEVICE
816
STORAGE DEVICE
818
MACHINE READABLE MEDIUM
826
INSTRUCTIONS
802
SIGNAL GENERATION DEVICE
820
SENSOR(S)
824

FIG. 8

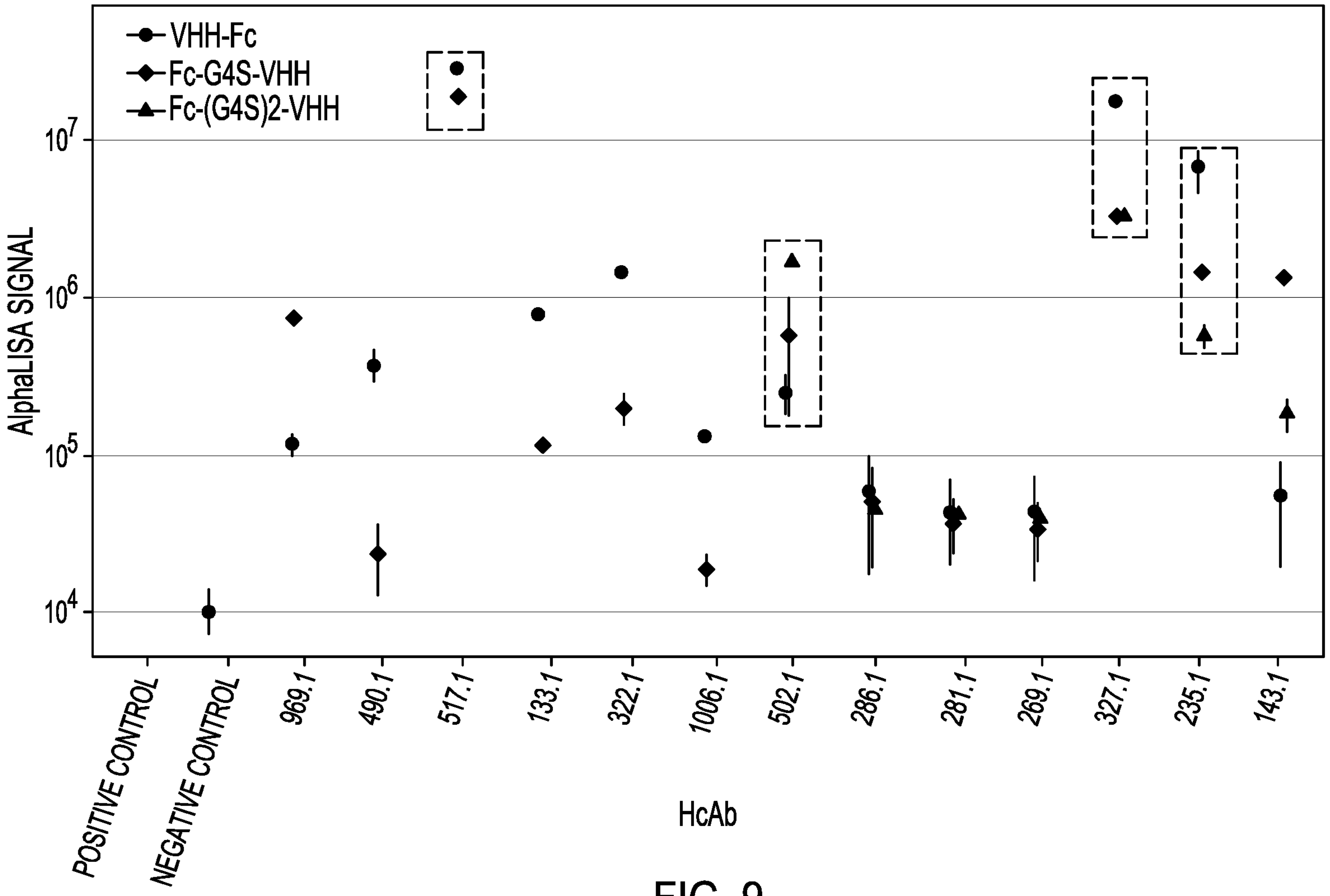
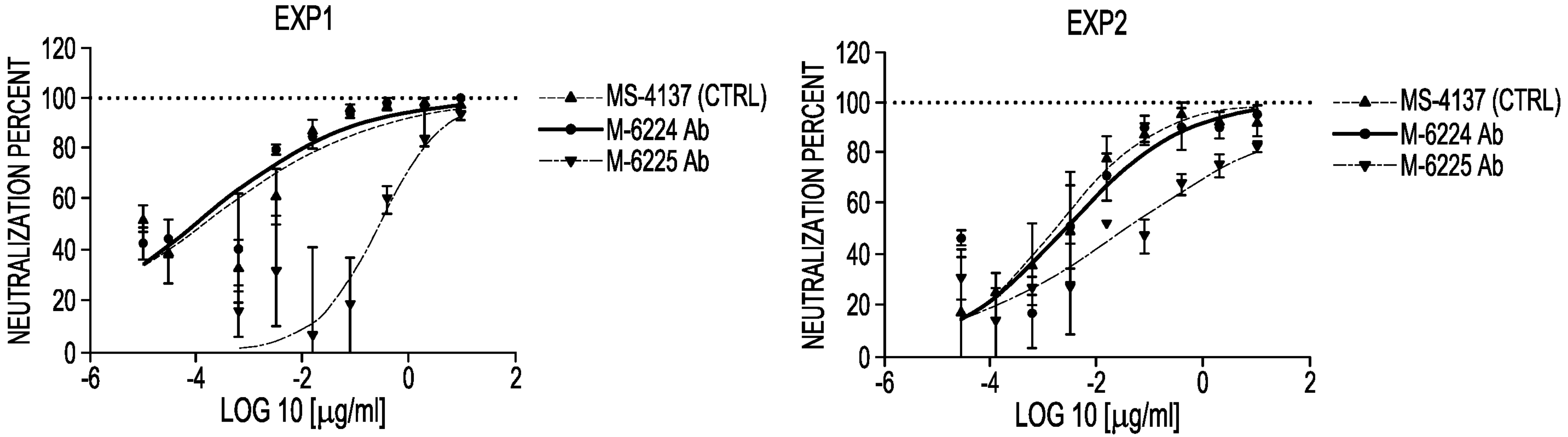

VHH-Fc
Fc-G4S-VHH
Fc-(G4S)2-VHH
AlphaLISA SIGNAL
10^7
10^6
10^5
10^4
POSITIVE CONTROL
NEGATIVE CONTROL
969.1
490.1
517.1
133.1
322.1
1006.1
502.1
286.1
281.1
269.1
327.1
235.1
143.1
HcAb

FIG. 9



	ESTIMATED ND50 μg/ml
ANTI-G (m102.4)	0,0003451
M-6224	0,0001037
M-6225	0,2693

	ESTIMATED ND50 μg/ml
ANTI-G (m102.4)	0,001655
M-6224	0,002559
M-6225	0,03450

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 23/34477

A. CLASSIFICATION OF SUBJECT MATTER

IPC - INV. G16B 40/00, G16B 20/00 (2023.01)
ADD. G16B 30/20, G16B 30/10, G16B 20/30 (2023.01)

CPC - INV. G16B 40/00, G16B 20/00

ADD. G16B 30/20, G16B 30/10, G16B 20/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2021/119472 A1 (Just-Evotec Biologics, Inc.) 17 June 2021 (17.06.2021) entire document (especially para [0052]-[0053], [0058]-[0060], [0068]-[0071], [0083]-[0084], [0095]-[0099], [00100]-[00104]).	1-21
A	US 2017/0206308 A1 (Yeda Research and Development Co. Ltd.,) 20 July 2017 (20.07.2017) entire document	1-21
A	US 2022/0093214 A1 (Thaumachron LLC) 24 March 2022 (24.03.2022) entire document	1-21
A	WO 2021/041199 A1 (Geaenzymes Co.) 04 March 2021 (04.03.2021) entire document	1-21

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 12 December 2023	Date of mailing of the international search report JAN 0 8 2024
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Kari Rodriquez Telephone No. PCT Helpdesk: 571-272-4300