SUPPLEMENTARY INFORMATION

MoleculeCrafter and Non-canonical Base Pairing: A Semi-automated CAD Tool for Creating Flexible and Unitized 3D Printable Macromolecules for Education

Elsa Schleicher[1] and Rebecca E. Taylor[12]*

[1] Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA

[2] Departments of Biomedical Engineering and Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

Corresponding author's email address: bex@andrew.cmu.edu

Table of Contents

Step-by-Step Tutorial for Model Kit Creation

Video Tutorials Available

Instructional video links can be accessed here:

- https://youtu.be/L3CvjZhtVsA,
- https://youtu.be/wjtzOpmGY3Q,
- https://youtu.be/pLvbBbAczyI,
- https://youtu.be/ZQIg3FLuI0Y,
- https://youtu.be/-2-jZup85Xw

Getting Molecular Data from a Database

Purpose: Here we will show a method for accessing molecular data that we can use to eventually make models in MoleculeCrafter. We show the RCSB Protein Data Bank here, but other sources may be used.

1. Search rcsb.org. This will bring you to the main page.
2. Type the PDB code of the molecule you want into the search bar.
    2.1. All molecules in the database are referred to by a 4-digit PDB code. If you read a paper with a particular molecule in it, it will often have its reference PDB code.
    2.2. For example, B-form DNA has the PDB code: 1BNA
3. You will see the main page for that molecule. It has lots of information including the paper which is the primary reference for the molecule.
4. Go to the Download Files icon in the right corner of the screen and scroll down to select PDBx/mmCIF format.
5. You are done! Check your downloads folder for the file.
    5.1. Note: for a quick way to see the data held in a file (before you even download it) go to the Display Files icon and select mmCIF Format
    5.2. This will open a new tab displaying all the data for that molecule. This step is not necessary for gathering data, but can be helpful for those who want more information on their downloaded file.

Viewing the File in ChimeraX

Adding Hydrogens in ChimeraX

Purpose: There are two types of files in the RCSB depending on how the structure in question was determined. Either a structure was determined by x-ray diffraction or by NMR. When we are discussing molecular models and their data, the determination of the structure is important because molecules determined by x-ray diffraction will sometimes not include the coordinates of hydrogen atoms, whereas molecules determined by NMR usually will have them. Since many bonds we will model are between hydrogen atoms and other atoms, if missing, we need to add the coordinates of hydrogens to the

molecular csv file we upload to MoleculeCrafter. There are many different ways to do this, but here we will present our method, which involves using ChimeraX[1].

1. Look at the RCSB page for your molecule of choice.
2. Next to "Method:" you will find the method by which the structure was determined.
    2.1. If the structure says X-RAY DIFFRACTION, then its data file is usually missing the coordinates of its hydrogen atoms and additional work needs to be done to add these in.
    2.2. If the structure says SOLUTION NMR, then its data file usually does contain hydrogen atom coordinates. However, we still recommend viewing the molecule in ChimeraX, as it can help you decide how you want to break up your molecule into unit pieces.
3. Open ChimeraX (we used v1.7.1). This is an open source tool and can be downloaded for free. For download instructions, see this link: https://www.cgl.ucsf.edu/chimerax/download.html
4. Select the Open icon in the top left corner of the screen.
5. Select the file downloaded in Part 1. And click Open.
6. ChimeraX allows you to change how you view your molecule. By default it uses "slabs" for nucleic/amino acids and "ribbons" for backbones. It is easier to see the addition of hydrogens by changing the visualization style.
7. Go to Nucleotides → select Plain.
8. Go to Molecule Display → Cartoons → select Hide
9. Still in Molecule Display → Styles → Ball and Stick
    9.1. Now you should be able to see all the atoms.
    9.2. So far you have just changed the visualization. You have not altered the data file in any way. We will do that next.
10. Go to Tools → Structure Editing → Add Hydrogens
11. Make sure the method you select is Steric Only. Then select OK.
    11.1. You should see atoms appear on the molecule. If you have followed the instructions here and have not changed any color preferences the atoms that appear should be white. These are the hydrogen atoms we were missing.
12. Now that the hydrogen atoms are added we can save this file with the coordinates of all the atoms. Go to Home → Save
13. Name your file. We suggest something that tells you it has been hydrogenated and also includes the PDB code of the original molecule.
14. Make sure the file type is still an mmCIF (*.cif, *.mmcif)
15. Select save.
    15.1. You now have a data file with all the coordinates you will need.

Planning in ChimeraX

Purpose: Determining how you want to break your molecule into unit parts is an important step of the model making process. How you split up your molecule is up to you. Just remember that the choices you make as you split up the molecule impact its final accuracy. For example, in our example molecule (1BNA) we have decided to break the molecule into 5 unique unit pieces. We will have a backbone piece, an adenine piece, a thymine piece, a guanine piece, and a cytosine piece.

You could print every individual subunit as its own unique part (e.g. print every adenine that shows up in 1BNA individually), but if there is a high degree of similarity between the instances of a particular subunit, you can choose a representative piece (e.g. print one adenine that represents all adenines in the 1BNA structure). Printing stacks of identical subunits makes models easier to design, print, and assemble, while losing some detail and accuracy. This is a tradeoff model makers should be aware of, and should drive the design of their own models.

For subunits with a high degree of similarity between their instances, you can choose a representative piece based on visual inspection or by using a bioinformatics tool such as DSSR[2] to extract the torsion angles and pseudotorsion angles for nucleic acids. This tool, which is the successor to 3DNA,[3] has also been incorporated into the Nucleic Acid Knowledge Base website listing for select nucleic acid structures. Users can therefore easily review the DSSR-annotated entries to help guide the determination of "representative" parts. Below we discuss using a visual inspection, since it highlights the thought process you should develop as you begin to think about your molecules in the context of breaking them up into reasonable subunits.

1. In ChimeraX go to Molecule Display → Coloring → nucleotide
    1.1. This changes the coloring to be color coded by nucleotide type.
    1.2. Blue = Thymine, Red = Adenine, Yellow = Cytosine, Green = Guanine
2. Start examining your structure from different angles. You should be looking for how the bonds between your different subunits look. This can change depending on what molecule you have. However, there are two main types of bonds to look for. (1) Hydrogen bonds (depicted with blue dashed lines across a gap between molecules. (2) Any other bonds (shown either by a "stick" in a ball and stick model or by where atoms overlap with each other).
3. Start by examining the hydrogen bonds between bases. Think about which ones seem to be the average orientation for your subunits across the molecule. Then consider any other bonds relevant to your molecule.
    3.1. Do you see any patterns?
    3.2. Are certain bonding pairs similar enough to each other that a representative piece would be appropriate here?
    3.3. How many individual unique parts do you need to adequately reflect this structure?
    3.4. Always consider the purpose of your models. What are you trying to showcase? Can you lose some accuracy in one area to better showcase another aspect of molecular interaction?
4. Hover over a particulate base or molecular segment with the mouse to get its name and location in its chain.
    4.1. For instance, hovering over a Cytosine could show something like: /B DC 15 C2
    4.2. This means this is Chain B, Residue Cytosine, Sequence number 15 in the chain, and the particular atom the mouse is hovering over is C2.
5. Record the chain letter, residue name, and strand sequence number for any residue you will want to model later.
6. Repeat steps 2-5 for any subunits you want. (e.g. I'd find the residue information for my representative cytosine, guanine, adenine, thymine, backbone for my 1BNA model).

7. To add connectors to the subunits we have chosen later on in MoleculeCrafter, we will need the location of the atoms on the partners to which the subunits will bond.
    7.1. So, if I choose a particular cytosine residue, I need to also mark down that particular cytosine's guanine partner.
    7.2. So, if I have chosen the backbone connected to DT 7 on Chain A, I will also need the coordinates for DT 8 on Chain A, and DT 6 on Chain A.
8. Repeat steps 2-5 for the partners of the subunits you want. (If the partner is not a subunit itself).
9. Now that you know what pieces you will need to extract coordinates for, we can move on to the next step.

Purpose: As noted above, while visual inspection is a great place to start the process of understanding your molecule, you can make even more accurate decisions by using data. For instance, the Nucleic Acid Knowledgebase (NAKB) contains DSSR (Dissecting the Spatial Structure of RNA) analysis for select molecules (this data is gathered using X3DNA-DSSR). Here you can find data regarding a molecule's torsional angles, glycosidic bond, sugar pucker, shear, stretch, stagger, etc. A model maker can use this information to help inform decisions regarding their chosen subunits.

1. Go to the NAKB: https://nakb.org/ (or browse the listing of DSSR-annoted NAKB entries here: https://nakb.org/nakblist=g4x3dna:*)
2. Type the PDB code of the molecule you want into the search bar.
    2.1. You may can also use its NDB code, which functions the same way
3. This will bring you to your molecule's main page.
4. Next to Analysis → hover of DSSR → select NAKB NA Parameters
5. You will now see the Nucleic Acid Parameters page.
6. Exploring these pages will provide information that can inform your design.
    6.1. For example, when making our 139D structure we checked the sugar pucker of the backbone and found them all to be C2' endo. So, we chose our representative backbone for 139D by choosing a specific residue from the molecule data to be our representative model.
    6.2. However, when making our 1BNA structure, checking the sugar pucker showed that the backbone can be C2' endo or C1' exo. We would need to choose a representative backbone carefully. Rather than choosing a C2' endo or C1' exo backbone to be a representative for both, we used a backbone designed from a calculation of the averages as reported in the original B-DNA dodecamer paper by Drew et al.[4]
7. We recommend leveraging the wide range of information available through the NAKB to make even more informed decisions about how to break up your molecule into subunits.

Parsing and Extracting the Atomic Coordinate Data

Purpose: To use MoleculeCrafter we need to extract the data from the mmCIF file and organize it into the appropriate format and then export it as a CSV. The CSV needs to be in the form of a 4-column table with one header row. The first column is for the x-coordinates, second column is for y-coordinates, third column is for z-coordinates, and the fourth column is the atomic radius of the atom. You may use your own tool to extract and manipulate your data, but to make this process easier we have made a Google

Colab notebook which walks users through the required steps and formats the data for them. This notebook uses our custom Python script, but does not require that the user have any knowledge or experience with Python. This Python script was made using the Biopython library,[5] which is often used for computational biology.

The Google Colab notebook, MoleculeCrafter Guided Parser, provides detailed instructions on how to use it. Sections A and B give background on how the parser works and also some important notes about using Google Colab. There are technically two parsers in this notebook: (1) the Standard Guided Parser, and (2) the Fast Residue Parser. The Standard Guided Parser is for first time users who want to understand how the parser works to break down the molecule. The Fast Residue Parser is much quicker and gives less guidance. It also does not handle typos well, and if something a user inputs is inaccurate, then it will stop running and the user will have to start again. Below we will walk the user through running each parser.

Getting Started

1. Go to the MoleculeCrafter Guided Parser: https://colab.research.google.com/drive/1l8c8AB8ofweAsAu5CAEN3BygvgQU1j-I?usp=drive_link#scrollTo=XSPfkbaD33x9
2. Make a copy of the parser for your own use.
3. Go to Section 1: Install Necessary Libraries
4. Hover your mouse over the square just next to the line that says !pip install biopython. A triangle facing right will appear. Click the triangle to run the code.
    4.1. Note that even if it looks like the command has been run before (either by you in a previous session or by us when making it), you will need to follow these steps. It is always better to re-run all commands when extracting your data to ensure there are no errors.
5. Wait for the command to run. When it is successful a green checkmark will appear just to the right of the code.
6. Go to Section 2: Upload File
7. Press the triangular play button next to the code block to run it.
8. A pop up at the bottom of the code block that says Choose File will appear. Click the Choose File button.
9. Select your file, click open.
10. Wait for the file to upload. When it is successful a green checkmark will appear just to the left of the code.
11. Go to Section 3: Desired PDB Code
12. In the text box type in the 4-digit PDB code of your molecule.
    12.1. An example PDB code may be in the text box already. Or it may be leftover from one of your own previous sessions. You will still need to type in the PDB code you want and run the command.
13. Press the triangular play button next to the code block.
14. Wait for the command to run. When it is successful a green checkmark will appear just to the left of the code.

15. Decide which parser you would like to use. For the Standard Guided Parser see steps 16-33 below. For the Fast Residue Parser skip to steps 39-45 below.

Using the Standard Guided Parser

16. Go to Section 4: Standard Guided Parser
    16.1. Note that if you click the (>) symbol or click Show Code in the code block for the parser (or any of the code blocks) it will show you the raw Python file used to extract the data. You do not need to look at the code to run the parser, but for any user who is interested in what commands are being run, they can look through it there.
17. Press the triangular play button to run the code.
18. You will be told what Entities are in your molecular file and given their respective IDs. You will be asked if you want to model any Entity or break the structure down more.
    18.1. If you want to model an entity, type the ID number of the entity into the text box and press enter/return button on your keyboard. Skip to step 33
19. To break the molecule down more, type YES in the text box and press enter. You must use all capital letters and do not use any extra spaces.
20. You will be told what Models are in your molecular file. You will be asked if you want to model any of the Models or break the structure down more.
    20.1. If you want to model a Model, type NO and press enter.
    20.2. Type the Model ID number of the Model you want to choose into the text box and press enter. Skip to step 33
21. To break the molecule down more, type YES in the text box and press enter.
22. You will be told what Chains are in your molecular file. You will be asked if you want to model any of the Chains or break the structure down more.
    22.1. If you want to model a Chain, type NO and press enter.
    22.2. Type the Chain ID letter of the Chain you want to choose into the text box and press enter. (Must be a capital letter).
    22.3. If there are multiple Models to choose from, type the Model ID number of the Model which contains the Chain you want into the text box and press enter. Skip to step 33
23. To break the molecule down more, type YES in the text box and press enter.
24. You will be told what Residues are in your molecular file. You will be asked which Residue you want to model.
25. Type the 2 or 3 letter Component ID of the Residue you wish to model and press enter. (i.e. DC for Cytosine).
26. You will be told that polymer Residues occur in a polymer chain sequence. You will be asked if you would like to be given any occurrence of the Residue you chose or if you have a particular sequence number in mind.
    26.1. Type YES and select enter if you do not have a sequence number and do not care which Cytosine is given. Skip to step 33
    26.2. Note, that this choice isn't particularly recommended for model makers. If you have completed Part 2: Planning in ChimeraX, then you should have analyzed your structure for the sequences you want. However, if your goal is just to practice extracting data from an mmCIF file and uploading it in MoleculeCrafter, this is a helpful command for those learning the ropes.

27. To give a specific sequence number, type NO and press enter.
28. Type the integer sequence number you want and press enter. (i.e. if I want DC 15, I type in 15)
29. If there are multiple Chains to choose from in your structure, type the Chain ID letter of the Chain you want to take your residue from into the text box and press enter. (Must be a capital letter). If there is only one Chain, you will not be asked to make a choice.
    29.1. Note, if your Residue does not exist at that sequence number on that chain, you will get an error message. Make sure you select the chain your residue is on. You should have planned this out in Part 2.
30. If there are multiple Models to choose from in your structure, type the Model ID number of the Model which contains the Residue you want into the text box and press enter. If there is only one Model, you will not be asked to make a choice.
31. You will be told what neighbor Residues were found in the vicinity of your chosen Residue. You will be asked if you want to export the neighbors as well as your chosen Residue or just your chosen Residue.
    31.1. Note: exporting neighbors is a quick way of getting the surrounding residues, which may be helpful when modeling connectors in MoleculeCrafter. However, it often finds a lot of extraneous neighbors which are unnecessary. The best thing to do is to have all the residues you want planned out ahead of time.
32. To just export your chosen Residue, type NO and press enter.
33. You will be told that your file has been exported and you will be told its file name.
    33.1. Note: it will also give a list of names for the neighbor files regardless of whether you chose to export them or not. If you choose not to export them, they won't show up.

Viewing Parsed File

34. Go to Section 6: Review Output CSV File
35. Go to the file icon (shaped like a file folder) on the left hand side of the screen.
36. Find the file with the title shown in Step 33. Click on it to view the file.
37. Close the tab once you have viewed the file.
38. You can reuse the Standard Guided Parser to get all of the CSVs you need for modeling.
    38.1. This does take time. The Fast Residue Parser is quicker (especially if you have planned out all the specific residues and their chain and sequence information ahead of time, as we showed in the ChimeraX section of these instructions)

Using the Fast Residue Parser

39. Go to Section 4: Fast Residue Parser
40. Press the triangular play button to run the code.
41. You will be told what Residues are in your model and asked which you would like to model.
    41.1. Type the 2 or 3 letter Component ID (i.e. DG)
    41.2. Must be all CAPITALS with no spaces
    41.3. Press enter
42. You will be asked for the sequence number.
    42.1. Type the numerical sequence number for your residue (i.e. 10)
    42.2. Press enter

43. You will be asked for the Chain ID.
    43.1. Type the CAPITAL letter for the chain (i.e. A)
    43.2. Press enter.
44. You will be told that your residue has been parsed and asked if you would like to parse another.
    44.1. Type YES to give another residue to export into an appropriate CSV.
        44.1.1. Repeat steps 41-44.
    44.2. Type NO if you are done entering residues and would like the ones you have already input to be exported.
    44.3. Hit enter.
45. You will be told that your file(s) have been exported and you will be told their file name(s).
46. See steps 34-37 to view these exported files.

Downloading your CSV

47. Finally, go to the file folder icon on the left to see your list of CSVs.
48. Right click on a CSV and click download to download them to your computer.
    48.1. You cannot leave the CSVs in Google Colab indefinitely. It does not keep data from past sessions.
    48.2. It is best to download all data immediately after parsing it.
49. Repeat steps 47-48 for all CSVs.

Creating Molecules and Adding Connectors

Purpose: A guide on how to use MoleculeCrafter's custom features to make models.

Creating Subunits for your Molecular Models

1. Open the MoleculeCrafter document workspace: https://cad.onshape.com/documents/32717bfbc9f5e2ecec50465b/w/3d86996cbfc1ad808af54aa9/e/2672fefd7baba827b5c4f509
2. Select, Make a Copy to Edit button in the top middle of the screen.
    2.1. You will now be able to make edits to your own MoleculeCrafter document workspace without altering the original public document.
3. Or you can go to the icon with the three horizontal bars in the top left corner of the screen and select Copy Workspace.
4. Once in your own document workspace, click on the CSVs folder in the bottom toolbar.
5. Click on the plus icon in the bottom left hand corner.
6. Select import.
7. Click on the CSV or CSVs of data you want to to import. Press open.
8. Dismiss import notifications by clicking the X on the top right of the notifications screen.
9. Check that all the CSV files are in the CSV folder.
    9.1. If they are not you can drag and drop them into the correct folder.
10. Click on the Begin Modeling Part Studio.

10.1. Note this is a default part studio. You can add additional part studios to build in by going to the plus icon and selecting Create Part Studio.

11. Go to the top right corner of the toolbar and find the custom feature menu. This icon looks like a cube in between {curly brackets}
12. Select Molecule Creator.
13. Click on Select CSV File.
14. Select your CSV file from the options that appear.
15. Press the green check mark to confirm your feature.
16. Scroll out/look around to find your generated molecule.
    16.1. It rarely generates in the center of your screen, since it will generate wherever its atomic coordinates define it. Some zoom control will be necessary.
17. Go to the custom feature menu.
18. Select Molecule Unitizer.
    18.1. Each sphere is currently one part. We want to combine them into the subunits we will eventually print as unique parts.
19. Click the atoms you want to join into one subunit.
    19.1. They will highlight yellow when selected.
    19.2. Click them a second time to deselect.
20. Choose a color to make the subunit from the dropdown color menu in the Molecule Unitizer options screen.
21. If you want to delete specific atoms, select the Delete Select Atoms check box. Then select any atoms you want to delete.
22. Click the green check mark on the Molecule Unitizer options screen to confirm your choices and complete your feature.
23. Your subunit should now be one part. Confirm this by selecting the part and looking at what becomes highlighted in the Parts List on the left bottom half of the screen.
24. It is often helpful to name your part something distinctive at this point. Right click on the part in the Parts List and select Rename.
    24.1. We recommend labeling the part by residue type, chain id, and sequence number.
25. Repeat steps 18-23 until you have made all the subunits you want out of this particular CSV file.
26. Repeat steps 11-24 for every CSV you wish to generate to make the subunits you need.
    26.1. Note, sometimes you will need to generate more parts than you will print, because you will use other pieces for aligning the custom connectors later.
    26.2. Note, you can hide certain components, which makes it easier to see and select other components. To hide a component, right click on the part and select Hide.
        26.2.1. Alternatively, hover over the name of the part in the Parts List on the left of the screen and click the eye icon to hide/unhide the part.
27. It's helpful to organize your features in the feature timeline into folders. Click on the folder icon with a plus on the top left hand corner of the screen. Name your folder something like Molecule Creators.
28. Click the green check mark.
29. Select all of the Molecule Creator and Unitizer custom features listed in the feature timeline and drag them into the newly created folder.

Adding Connectors

Purpose: Explains how to add the different kinds of connectors and recommendations for where they go. Note that you can decide to put a connector in a different location depending on the subunits you wanted to divide your molecule into. These are our recommendations for starting points and are based on teaching nucleic acid bonding.

Adding Pin Connectors

Pin Connectors: These are typically used to represent hydrogen bonds. In our DNA and RNA centric models they are used for bonds between bases along the Watson-Crick-Franklin, Hoogsteen, and Sugar Edges of the nucleic acid bases.

1. Go to the custom features menu.
2. Click Pin Connector.
    2.1. Note, always do the socket connectors second, since the creation of the socket makes it difficult to select the center of the atom for aligning its partner connector.
3. Choose the origin of the connector. This is the atom on which the connector originates.
    3.1. Hover over the atom of choice until a Mate Connector appears (this looks like a circle with an XYZ axis inside it)
    3.2. Select the Mate Connector at the center of the atom.
    3.3. Note that Pin Connectors are designed to always go on hydrogen atoms. Unless you are doing something special, these connectors should originate on the hydrogen atoms in a hydrogen bond.
4. Now choose the connector alignment point. This is the center of the atom which will hold the partner connector of our Pin Connector (which is the Pin Socket).
    4.1. Hover over the atom of choice until a Mate Connector appears.
    4.2. Select the Mate Connector at the center of the atom.
    4.3. Note, this alignment atom will usually be something like a Carbon, Nitrogen, or Oxygen atom. These are the atoms which will eventually hold the partner connector of Pin Socket.
5. Select the object to join the connector to. This should be the same part that contains the origin for the connector.
    5.1. Click the face of the object to which the connector is being added.
6. Now that all the fields in the Pin Connector options screen are filled out, the connector should appear as a translucent potential part on your model.
7. Select the arrow icon next to the Object to Join field to switch the direction of the connector piece, if necessary.
8. The Truncated Pin Connector option is for cases when you will have two (or more) hydrogen bonds that bond with the same partner atom on another subunit. Currently, there is no way to make two Pin Connectors go to one socket on the same single atom. So, you can truncate one of the two Pin Connectors. This means it will not contain the part which physically clicks into a socket, but there will be a shortened cylindrical piece attached to the hydrogen atom to show that that particular hydrogen is meant to be forming a bond.
    8.1. To create a Truncated Pin Connector, select the Truncated Pin check box.

9. Click the green check mark to confirm your options and create your Pin Connector.
10. Repeat steps 1-9 to add all Pin Connectors you need to your models.

11. Go to the custom features menu.
12. Click Pin Socket.
    12.1. Note, always do the socket connectors second, since the creation of the socket makes it difficult to select the center of the atom for aligning its partner connector.
13. Choose the origin of the connector. This is the atom on which the connector originates.
    13.1. Hover over the atom of choice until a Mate Connector appears
    13.2. Select the Mate Connector at the center of the atom.
    13.3. Note that Pin Sockets are not designed to fit on hydrogen atoms. Unless you are doing something special, these connectors should originate on Carbon, Nitrogen, Oxygen, or similar atoms.
14. Now choose the connector alignment point. This is the center of the atom which will hold the partner connector of our Pin Socket (which is the Pin Connector).
    14.1. Hover over the atom of choice until a Mate Connector appears.
    14.2. Select the Mate Connector at the center of the atom.
    14.3. Note, this alignment atom should be a hydrogen atom.
15. Select the object to join the connector to. This should be the same part that contains the origin for the connector.
    15.1. Click the face of the object to which the connector is being added.
16. Now that all the fields in the Pin Socket options screen are filled out, the connector should appear as a translucent potential part on your model.
17. Select the arrow icon next to the Object to Join field to switch the direction of the connector piece, if necessary.
18. Change the rotation angle to change where the slot in the Pin Socket cuts into the atom.
    18.1. Type in an angle value.
        18.1.1. It is best to use trial and error to find a good angle value. It is often not consistent across atoms, so we don't have a recommended angle to give. Start with larger angles and then make smaller guesses once you get close to what looks best.
        18.1.2. Ideally, we want a slot that cuts into the atom which contains the Pin Socket and not the atoms neighboring it.
    18.2. Alternatively, you could hover over the atom containing the Pin Socket and drag the arrow icon that appears.
        18.2.1. We don't recommend this option as it can be very touchy.
19. Choose the cut length of the slot.
    19.1. Select the cut you want from the dropdown menu.
    19.2. These different cut lengths affect how open/closed the Pin Socket is, and thus affect how loose the Pin connection joint will be in your model.
    19.3. We highly recommend the default cut, but have several options here in case your printing facility of choice requires a different level of precision.
20. Click the green check mark to confirm your options and create your Pin Socket.
21. Repeat steps 11-20 to add all Pin Sockets you need to your models.

Adding Duckbill Connectors

Duckbill Connectors: These connectors have one degree of rotational freedom. These are typically used to connect nucleic acid bases to sugar-phosphate backbones. However, you can use them in other places that would be helpful to your specific purposes. They are made to fit on atoms like Carbon, Nitrogen, Oxygen, etc. They will not fit on hydrogen. These connectors are harder to fit together than the Pin connectors. Pins can more easily snap in and out, duckbill connectors are meant to attach and then allow rotation, but not come apart as easily.

1. Go to the custom features menu.
2. Click Duckbill Connector.
    2.1. Always do the socket connectors second.
3. Choose the origin of the connector. This is the atom on which the connector originates.
    3.1. Hover over the atom of choice until a Mate Connector appears.
    3.2. Select the Mate Connector at the center of the atom.
4. Now choose the connector alignment point. This is the center of the atom which will hold the partner connector of our Duckbill Connector (which is the Duckbill Socket).
    4.1. Hover over the atom of choice until a Mate Connector appears.
    4.2. Select the Mate Connector at the center of the atom.
5. Select the object to join the connector to. This should be the same part that contains the origin for the connector.
    5.1. Click the face of the object to which the connector is being added.
6. Now that all the fields in the Duckbill Connector options screen are filled out, the connector should appear as a translucent potential part on your model.
7. The neighbor atom which will hold the Duckbill Socket will often obscure the connector.
    7.1. Hide the part which holds the connector alignment point.
8. Select the arrow icon next to the Object to Join field to switch the direction of the connector piece, if necessary.
9. Change the rotation angle to change where the slot in Duckbill Connector is located relative to the rest of the part.
    9.1. Type in an angle value.
    9.2. Or drag the arrow.
    9.3. This is purely aesthetic and does not really affect the ability of the connector.
10. Click the green check mark to confirm your options and create your Duckbill Connector.
11. Repeat steps 11-20 to add all Duckbill Connectors you need to your models.

12. Go to the custom features menu.
13. Click Duckbill Socket.
    13.1. Always do the socket connectors second.
14. Choose the origin of the connector. This is the atom on which the connector originates.
    14.1. Hover over the atom of choice until a Mate Connector appears.
    14.2. Select the Mate Connector at the center of the atom.
15. Now choose the connector alignment point. This is the center of the atom which will hold the partner connector of our Duckbill Socket (which is the Duckbill Connector).
    15.1. Hover over the atom of choice until a Mate Connector appears.

15.2. Select the Mate Connector at the center of the atom.

16. Select the object to join the connector to. This should be the same part that contains the origin for the connector.
    16.1. Click the face of the object to which the connector is being added.
17. Now that all the fields in the Duckbill Socket options screen are filled out, the connector should appear as a translucent potential part on your model.
18. The neighbor atom which will hold the Duckbill Connector will often obscure the connector.
    18.1. Hide the part which holds the connector alignment point.
19. Select the arrow icon next to the Object to Join field to switch the direction of the connector piece, if necessary.
20. Click the green check mark to confirm your options and create your Duckbill Socket.
21. Repeat steps 11-20 to add all Duckbill Sockets you need to your models.
    21.1. Note that if you are using just one backbone as your average subunit, you really only need to model the Duckbill Socket on the one backbone you plan to use as your average subunit.

Adding Ball Connectors

Ball Connectors: These connectors have three degrees of rotational freedom. These are typically used to connect sugar-phosphate backbones to other sugar-phosphate backbones. However, you can use them in other places that would be helpful to your specific purposes. They are made to fit on atoms like Carbon, Nitrogen, Oxygen, etc. They will not fit on hydrogen. The Ball Socket will go on the backbone end with a central atom and three surrounding atoms (the 5' end, this is the phosphate group). The Ball Connector will go on the opposite end, which looks like a single atom (3' end).

1. Note, an easy way to see where one nucleic acid residue's backbone unit begins and ends (when looking at a chain of multiple backbone parts) is to click on the backbone part in question. It will highlight yellow, easily identifying it in the chain.
2. Go to the custom features menu.
3. Click Ball Connector.
    3.1. Always do the socket connectors second.
4. Choose the origin of the connector. This is the atom on which the connector originates.
    4.1. Hover over the atom of choice until a Mate Connector appears.
    4.2. Select the Mate Connector at the center of the atom.
5. Now choose the connector alignment point. This is the center of the atom which will hold the partner connector of our Ball Connector (which is the Ball Socket).
    5.1. Hover over the atom of choice until a Mate Connector appears.
    5.2. Select the Mate Connector at the center of the atom.
6. Select the object to join the connector to. This should be the same part that contains the origin for the connector.
    6.1. Click the face of the object to which the connector is being added.
7. Now that all the fields in the Ball Connector options screen are filled out, the connector should appear as a translucent potential part on your model.
8. The neighbor atom which will hold the Ball Socket will often obscure the connector.
    8.1. Hide the part which holds the connector alignment point.

9. Select the arrow icon next to the Object to Join field to switch the direction of the connector piece, if necessary.
10. Click the green check mark to confirm your options and create your Ball Connector.
11. Repeat steps 2-10 to add all Ball Connectors you need to your models.

12. Go to the custom features menu.
13. Click Ball Socket.
    13.1. Always do the socket connectors second.
14. Choose the origin of the connector. This is the atom on which the connector originates.
    14.1. Hover over the atom of choice until a Mate Connector appears.
    14.2. Select the Mate Connector at the center of the atom.
15. Now choose the connector alignment point. This is the center of the atom which will hold the partner connector of our Ball Socket (which is the Ball Connector).
    15.1. Hover over the atom of choice until a Mate Connector appears.
    15.2. Select the Mate Connector at the center of the atom.
16. Select the object to join the connector to. This should be the same part that contains the origin for the connector.
    16.1. Click the face of the object to which the connector is being added.
    16.2. NOTE: For this connector only, YOU CAN NOT CHOOSE THE FACE OF THE ATOM TO WHICH YOU ARE ADDING THE BALL CONNECTOR. The connector will not appear. This has to do with the way the custom feature is programmed to add the connector.
    16.3. Choose any face on the backbone which is not the face of the atom which contains the origin of the connector.
17. Now that all the fields in the Ball Socket options screen are filled out, the connector should appear as a translucent potential part on your model.
18. Change the rotation angle to change where the slots in Ball Socket cut into the atom
    18.1. Type in an angle value.
    18.2. Or drag the arrow.
    18.3. The goal is to align the slots so that they only cut into the central atom which holds the connector, not the neighboring atoms.
19. Select the arrow icon next to the Object to Join field to switch the direction of the connector piece, if necessary.
20. Click the green check mark to confirm your options and create your Ball Socket.
21. Repeat steps 12-20 to add all Ball Sockets you need to your models.

Scaling Parts

Purpose: Parts are created at a much larger scale than you would want to print at. They must be scaled before printing. You can scale your parts externally (outside of Onshape) if you have a preferred method. Alternatively, you can use Onshape's standard scaling tool. Or, you can use MoleculeCrafter's Custom Scaler feature, which we describe here. The scaler works by scaling the parts based on what final radius the hydrogen atoms should be (in mm).

1. Click the plus icon in the bottom left hand corner of your document workspace.
2. Select Create New Part Studio.
3. Rename it something like Stacks.
4. Select the arrow next to the Plane icon in the toolbar at the top of the page. This will open a dropdown menu of more options.
5. Scroll to the bottom of the dropdown menu and select derived.
6. Click Select Part Studio → Begin Modeling → Choose the subunit(s) you want to print.
7. Dismiss the selection window by pressing the X in its top right hand corner.
8. Select the green check mark.
9. Go to the custom features menu.
10. Click Custom Scaler.
11. Click objects to scale.
12. Select the faces of all objects you want to scale.
13. Choose the scale factor from the dropdown menu.
    - 13.1. The 100% (1mm:0.87Angstroms) and 75% (1mm:0.65Angstroms) scale are our default options and are named based on our past model creations and paper.[6]
    - 13.2. Select Choose Custom option to scale your parts to your own desired size.
    - 13.3. Hover over the Final Atomic Radius field to see information regarding scaling. It will tell you what the radius of hydrogen is right now (7.5mm) and what radius it is for 100% and 75% scales (1.69mm, 1.27mm).
    - 13.4. Type in the value (in mm) you want the final atomic radius of the hydrogen atoms to be.
14. Click the green check mark to confirm your options and create your Ball Socket.

Aligning and Stacking Parts

Align Parts for Stacking

Purpose: It is easier to stack parts once they are aligned with the top plane. Since parts are imported based on atomic coordinates, they are always at unusual angles. The Pre-Stack Aligner makes it easier to manipulate parts than it would be to move them around by hand using Transforms, rotations, translations, etc.

1. Click the sketch icon in the top left hand corner of the toolbar.
2. Select the top plane as the sketch plane.
3. Click the Top side of the cube in the right hand corner of the screen to rotate the screen to be a top down view of the top plane.
4. Click the point tool on the toolbar.
5. Make a point for each of your subunits horizontally in line with the origin.
6. Click the dimension tool.
    - 6.1. Click the first point and then the origin. Type in a distance for them to be.
        - 6.1.1. They should be at least twice as wide as your subunits.
    - 6.2. Dimension all the points in this same way.
7. Click the green check mark to exit the sketch.
8. Go to the custom features menu.

9. Click Pre-Stack Aligner.
10. Click 3 mate connector points in your subunit that you want to align with the top plane.
    10.1. This means that the three points you choose in your subunit will all intersect (or be parallel with) the top plane.
11. Click the face of the object you are aligning.
12. By default this part will be moved to the origin.
    12.1. Click Move to a Specific Location checkbox to define a location for the part to go.
    12.2. Click on one of the points of the sketch you made previously. This will move the part to that location.
13. By default this action will create mate connectors at the part's final location which will face in the Z and X direction.
    13.1. You will use these to stack later. Usually you only need the default.
14. By default the distance the part will be from the top plane is 0mm.
    14.1. Type a value in mm to offset the part a certain distance from the top plane.
    14.2. You will only need to do this if you are making a stack from parts that are not all identical. For example, if you had two types of Cytosine in your molecule (one which bonds by Hoogsteen bonds and one which bonds by Watson-Crick-Franklin bonds). You could align one with a point and then align another with the same point, and then offset the second part so that it is above the first part and does not intersect with it.
15. Click the green check mark.
16. Repeat steps 9-15 to align all subunits.

Stack Parts to Print Identical Pieces

17. Click the linear pattern feature in the main toolbar at the top of the page (tool icon looks like a pattern of four cubes in grid).
18. For Entities to Pattern, select the first subunit you want to make into a stack.
19. For Direction, select one of the mate connectors created by the Pre-Stack Aligner.
    19.1. Typically choose the one perpendicular to the Top plane.
20. Change the distance to a mm value such that the patterned pieces do not quite touch each other.
    20.1. How far apart you make them depends on the quality of the service you use for printing. Printing things too close together can cause printing errors with the connectors (especially sockets).
    20.2. When in doubt, spread the pieces further apart.
21. Change the instance count to the number of duplicate pieces you want.
    21.1. The number of units you have per stack is up to you.
    21.2. Remember not to make a stack excessively tall, because then it will not fit on a printing bed that well. We often print stacks of 16.
    21.3. You can always print more than one stack to get all the pieces for any given model.
22. Click the green check mark.
23. Orient the screen so you are looking at your parts top down (looking at the Top plane). Click the view cube in the right hand corner to change your orientation.
24. Select sketch.
25. Select the Top plane as the sketch plane.
26. Click the circle tool from the top toolbar.

27. Draw a circle on one of the larger atoms in your stack (C, N, O, etc)
    27.1. The circle should be in the center of the atom, if possible.
    27.2. Choose an atom that does not hold a connector.
        27.2.1. Technically, it could hole a connector, but to avoid interference in printing it's best to keep this part of the stack away from important features of the molecule so that it does not obscure them.
28. Use the dimension tool to size the circle.
    28.1. Make it large enough to be printed and not snap immediately, but small enough that you don't waste material and can break it when you twist the parts away from each other after the stacks are printed.
    28.2. 1mm or so is reasonable for the 75% scale.
29. Click the green check mark.
30. Click the extrude feature in the standard tool bar (icon looks like a gray cube over a white cube).
31. For faces and sketch regions to extrude, select the sketch we just made.
32. Drag the arrow on the model upwards until the circle (now a cylinder) extrudes through all of the parts in the stack.
33. At the top of the Extrude options popup, click Add (the default is New)
34. The Merge scope field will appear.
35. Click the Merge with all checkbox to make all the parts (and the extruded cylinder) one complete part.
36. Click the green check mark.
37. Repeat steps 17-36 for the other stacks you want to complete.
38. Note, subunits that are not as nicely planar as nucleic acid bases (such as the sugar-phosphate backbones) can be stacked with the same steps as above with a few small tweaks.

39. Repeat steps 17-26 for stacking backbones.
    39.1. Keep more space between the backbones than the bases.
    39.2. You really don't want any of the connector joints to be close to each other. If they are too close they can merge together during printing and obscure key features.
    39.3. How far apart they should be depends on your printing vendor of choice.
40. When you get to the step where you draw a circle, do not draw the circle on an atom.
41. Instead, choose an atom on the outside away from any connectors. Draw a circle that partially intersects this atom that sticks out.
    41.1. The overlap need not be very large. In fact, you want it to be quite small, so it is easy to break the individual pieces off after printing.
42. Dimension the circle.
    42.1. It will need to be a bit bigger than the one for the bases. This cylinder need not be broken to get the individual connectors. Instead the connectors will be twisted off.
43. Repeat steps 29-36.
    43.1. Often when extruding through all the parts, you will need to click the Second end position check box.
    43.2. Then drag the second arrow in the opposite direction of the first to extend the cylinder below the top plane if necessary.
44. Click on the arrow next to the linear pattern tool → click circular pattern

45. For entities to pattern, click the backbone (or non-planar subunit) stack.
46. For axis of pattern, click the circular edge at the top (or bottom) of the cylinder.
47. You should see the grayed out example of what your model will look like with this pattern.
48. Change the view to top down to see how the part looks circling that center cylinder.
49. Change the instance count so that the subunits do not intersect each other and only intersect the central cylinder at one (small) point.
    49.1. Typically you would use 2 or 3 instances.
50. Go to Add → Merge scope → select the circular patterned pieces that should be merged together.
51. Click the green check mark.
52. Check that the merge worked properly and the stack is all one part.

53. Find the stack in the part list on the left side of the screen.
54. Right click the part name in the part list → select export
55. Rename your file.
56. Select an export format
    56.1. Recommend STEP or STL
57. Click export
58. Repeat steps 53-58 for all stacks.
59. Now you can print your parts at a vendor of your choice.

A Detailed Feature Guide for MoleculeCrafter

Creating a methodology for generating modular flexible molecular models required a computer-aided design (CAD) software system which would facilitate a method for semi-automating the affixing of connectors to part molecules, as well as resources to source, visualize, parse, and extract the requisite molecular data. The following provides an overview of the chosen CAD software platform, Onshape, and how its unique properties enabled the creation of MoleculeCrafter. This document also provides detailed information on how to perform initial steps A-E of Fig. 1 that are essential for gathering and processing molecular data for use with the Onshape-based tools. Finally, we provide links to all of the described tools and video tutorials of each step in the process.

About Onshape Accessibility and Customization

Onshape (PTC, Boston, MA) is a cloud-native CAD software platform that is accessible via web browser and is free for students, educators, and non-commercial use.[7] It does not require installation, and it functions similarly to traditional CAD systems for solid modeling in 3D, however, with the addition of a custom scripting tool called *Featurescript*, Onshape allows for the creation of custom features within the Onshape platform.

In a typical CAD system, a *feature* would be something like the *extrude* tool. Accordingly, a custom feature allows an Onshape user to define a custom tool which takes specific inputs to perform a non-standard action. Via FeatureScript, we designed a series of custom features to allow users to generate molecules from atomic coordinate data, group the subsequent atoms into appropriate units, and add various types of connectors, all without requiring the user to completely recreate the connector shapes at each desired instance on their molecule.

Onshape functions by creating a top-level workspace called a *document*. Document workspaces are made up of a number of pages which can hold part studios, assemblies, feature studios, drawings, and additional data such as CSV and STEP files. These pages appear as tabs at the bottom of the workspace, between which a user can navigate. The tabs in a document workspace can be arranged into folders for organizational ease. A *part studio* is a page where users can create a part, and it functions similarly to other CAD systems for solid modeling in 3D. All programming done in Onshape is completed in a *feature studio* page. Once a feature studio is created, the custom feature can be implemented in any part studio within the document workspace. The custom feature appears as an option in the main toolbar within a part studio and can be used as an additional tool in part modeling.

The tool we call MoleculeCrafter is a single Onshape document workspace which contains the following folders: *Connector Part Studios*, *Feature Studios Code*, *Icon SVGs*, and *CSVs*, as well as an initial part studio titled, *Begin Modeling*. *Connector Part Studios* contains part studios which reference the adding and cutting tool parts used to create the connectors. See the section

below in Adding/Cutting Tools for more details. *Feature Studios Code* contains all the feature studios that define the custom features listed in Table 1. *Icon SVGs* are reference images used to help identify the custom feature in the feature toolbar. *CSVs* is an empty folder to which a user is meant to upload their molecule's data. To utilize MoleculeCrafter, a user would make a copy of our document workspace. Once copied, they would be able to edit the workspace by uploading the CSV files of the desired molecule's atomic coordinate data. Next, the user would open the *Begin Modeling* part studio. Here they could begin implementing the custom features which facilitate the generation of their desired modular and flexible molecular model.

The joint types in Table 1 in the main text are adapted from the connector design for flexible DNA and peptide nucleic acid (PNA) models by Goodwin-Schoen and Taylor.[8] However, when combined with the MoleculeCrafter custom features, these connector features can be rapidly implemented on the molecule without requiring laborious CAD modeling of the joint (Fig. S1).

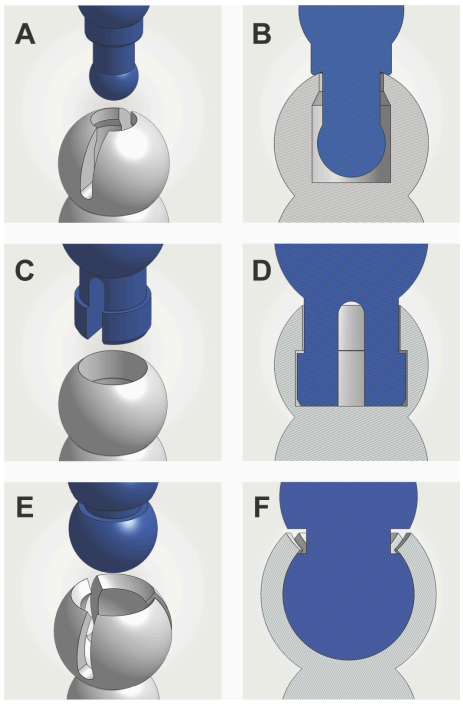

Figure S1. A summary of the joints adapted with permission from Goodwin-Schoen et al.[8] (A) Pin connector and pin socket joint. (B) Cross section of pin and pin socket. (C) Split-pin "duckbill" connector and duckbill socket joint. (D) Cross section of duckbill and duckbill socket. (E) Ball connector and ball socket joint. (F) Cross section of ball and socket.

Using Adding/ Cutting Tools to Make Connectors

The ease of connector incorporation was achieved by leveraging FeatureScript's ability to join and subtract previously created parts. To this end, a CAD part for each of the six connector types was created. These parts themselves then become either adding, or cutting tools to either add on to an atom or to cut away from it. Consider Figs. S2 and S3 below which illustrate the methods of adding and cutting, respectively.

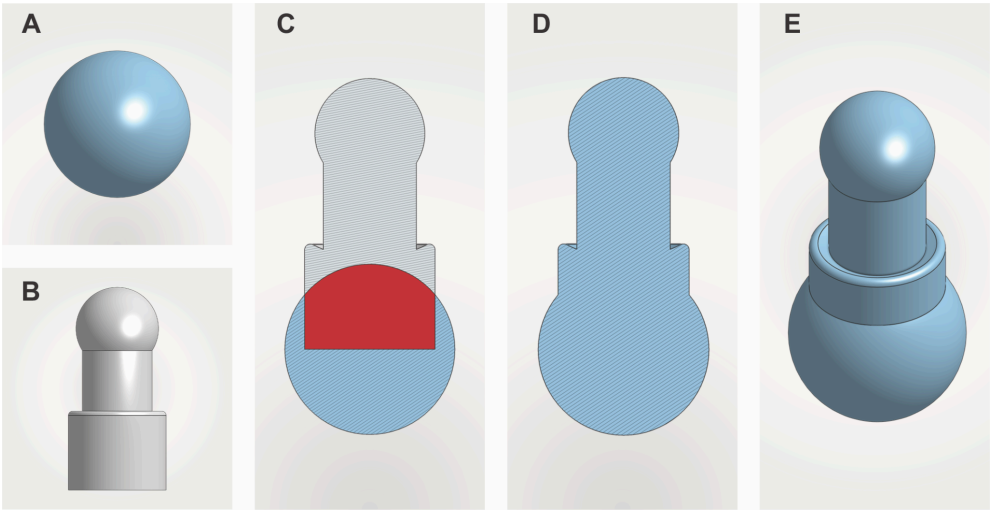

Figure S2. Method for adding a connector to an atom. Pin connector is shown as an example. (A) Spherical atom part to which a joint will be added. This is the target. (B) Previously designed pin connector part. This is the adding tool. (C) Cross-section showing the overlap of the two parts once the adding tool is instantiated on the atom part. (D) Cross-section of the final part after adding tool part and target part are merged. (E) Isometric view of final atom part with connector.

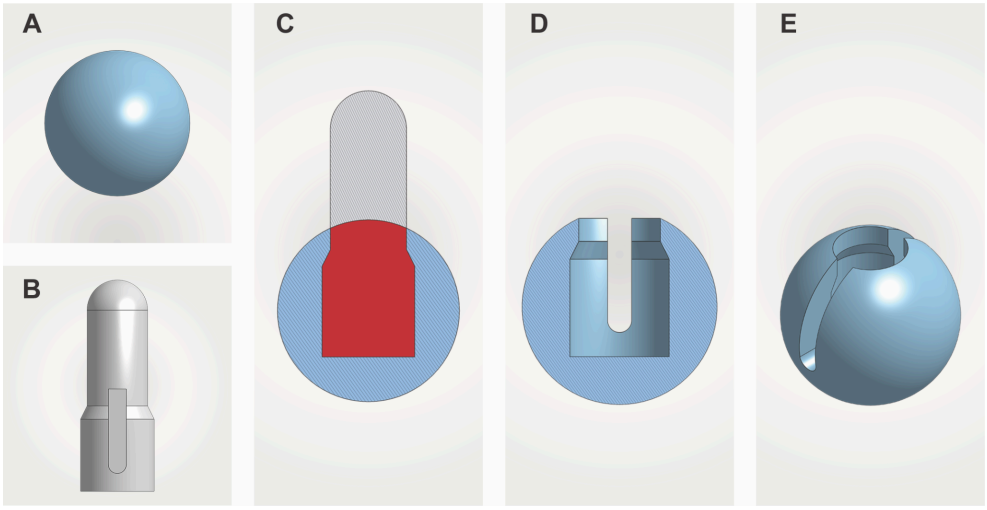

Figure S3. Method for using a cutting tool to create a joint feature. Pin socket is shown as an example. (A) Spherical atom part to which a joint will be added. This is the target. (B) Previously designed pin socket part. This is the cutting tool. (C) Cross-section showing the overlap of the two parts once the cutting tool is instantiated on the atom. (D) Cross-section of the final part after cutting tool removes the overlapping area. (E) Isometric view of final atom part with socket joint cut out of it.

As evidenced by Fig. S2, the Pin Connector tool instantiates a solid pin feature on the desired atom. The two pieces are then merged into a single part, thus creating the pin connector joint. Likewise, Fig. S3 shows how the cutting tool of the Pin Socket part is instantiated in the same manner, but when cutting, the overlapping region of the atom sphere part and the pin socket part is subtracted, leaving the desired pin socket joint. To this end, the connectors in Table 1 were made with adding tools, and the sockets with cutting tools.

Using this approach, the adding and cutting tool part for each connector type needed to be created only once; the connector custom feature thereafter references the adding/cutting tool to affix the specified custom connector to any desired atom or molecule. The adding and cutting tool parts remain as reference part files within the MoleculeCrafter document workspace. This means that if a user wanted to change some aspect of the connector design, they could edit the adding or cutting tool part directly. This works much like editing any CAD part and will automatically update so that the custom feature will reference the newly edited part, without requiring any alteration of the FeatureScript code. This attribute makes improving and testing MoleculeCrafter's connector design incredibly straightforward. A user who wished to add to or improve our design need only have proficiency in 3D CAD solid modeling rather than experience in FeatureScript programming.

CSV Upload and Integration

Since MoleculeCrafter leverages previously created parts, the comma separated value (CSV) file of atomic coordinate data and radii must be imported at the appropriate scale to interface with these parts. Onshape makes importing and accessing atomic coordinate CSVs a straightforward process. Onshape allows users to upload external data files, such as CSVs, to the document workspace where they can be referenced by pages in that workspace. FeatureScript compounds this power by enabling us to design MoleculeCrafter such that its custom features can reference an uploaded CSV as a tool input without requiring any direct editing of the FeatureScript code. Much like how the standard extrude tool requires a user to select a sketch as an input, MoleculeCrafter's custom feature, Molecule Creator, can take a user input selection of a previously uploaded CSV file.

Scaling the models

The Molecule Creator custom feature requires a CSV file in a specific format since it references previously created parts, which themselves are of a certain scale. It takes in a table of values with four columns and one header. The first three columns refer to the atomic X, Y, and Z coordinates, respectively. The fourth column gives the atomic radius of the atom. Table S1 indicates the list of radii values used to create the CSV files for the four structures. Additionally,

Table S1 identifies the equation used to determine the Final Atomic Radius (FAR) of any atom, should one not appear in the list below.

Table S1
Final atomic radii in mm for common biomolecules to be used with the Onshape custom features. The Consistent van der Waals radii are reprinted from Mantina et al. with permission from the American Chemical Society.[9]

Atom Type	Atomic Symbol	Consistent van der Waals Radius (Å)	68% VDW Radius (Å)	Final Atomic Radius (mm)
			Rounded to 2 significant digits	Multiplied by a factor of 10
Hydrogen	H	1.10	0.75	7.5
Carbon	C	1.70	1.2	12
Nitrogen	N	1.55	1.1	11
Oxygen	O	1.52	1.0	10
Phosphorus	P	1.80	1.2	12
Other	(-)	1.55	1.1	11

Equation for determining Final Atomic Radius (FAR) in mm

FAR = (Consistent VDW Radius) • 0.68 • 10

Table S1 illustrates that the atomic coordinates, originally in Angstroms, are later treated as millimeters once a user takes 68% of the consistent van der Waals (VDW) radius and multiplies by a factor of 10 to get the appropriate FAR value. The FAR value is the radius required for each atom in your CSV so that when the molecule is generated in Onshape, the atoms are at an appropriate size to interface with the pre-designed connectors. This calculation is done for you if you use our Standard Guided Parser to make your CSV file.

Later, you will want to scale these models down to an appropriate size for printing. If the user wants to use the Custom Scaler tool mentioned in Table 1, know that it scales the final structure assuming that the model's hydrogen atoms have a 7.5mm radius. (In other words it assumes you have assigned the atoms in your CSV with atomic radii according to Table S1). Prior to printing, the Custom Scaler tool can be used to shrink the models to a desired size.

Stacking the models for 3D printing

Once scaled, the Pre-stack Aligner tool, seen in Table 1, enables the user to reorient the desired piece and position it for stacking. Using the standard Onshape linear pattern tool will allow users to group the stacks into printable units with set numbers of repeatable parts. Advanced use of the Pre-stack Aligner can enable users to stack more than one type of piece in one stack. For instance, the stack of adenines for the RNA structure PDB-1DUL contained 5 distinct adenines. An additional custom tool, the Label Maker, allows the user to separate any distinct nucleic acid

bases in a stack with a label which contains the letter of the base, its bonding partner, and the geometrical nomenclature symbols for their respective bonding. Taken together, MoleculeCrafter's catalog of custom features facilitate every aspect of the generation of modular, articulated molecular models for 3D printing.

Creating a CSV with atomic radii and coordinates

In addition to Onshape our workflow requires resources to source, visualize, parse and extract the requisite molecular data as mentioned in steps A-E of Fig. 1. Specifically, MoleculeCrafter requires a CSV file containing atomic coordinates and atomic radii; how a user generates this CSV is up to the discretion of the user. The workflow and resources used to create the structural models presented in this paper are as follows.

Data Acquisition

The raw data for all molecules shown in this paper were taken from the *RCSB Protein Data Bank* (PDB) which is an online repository for experimentally-determined 3D structures and computed structure models made available for use in science and education.[10] A user can search for a specific molecule to model using its 4 digit alphanumeric PDB code. The data must be downloaded in a PDBx/mmCIF file format (which replaces the legacy PDB file format users may be familiar with).[11]

Note that other databases can also be used to find files. For instance, some papers may reference a molecule by its Nucleic Acid Database (NDB) code, which may make it necessary to look up via the Nucleic Acid Knowledgebase (NAKB).[12] While it is possible to download an mmCIF file from the NAKB, our experimentation shows that our custom Biopython script can sometimes give some errors using this method. To avoid this error, we recommend users look up the NDB file on the NAKB and find the corresponding PDB code for that same molecule. Using that PDB code in the RCSB Protein Data Bank will return the appropriate file that may be used with our custom parser.

Molecular Viewing and Hydrogen Addition

The majority of molecular structural files in the RCSB Protein Data Bank are determined via x-ray crystallographic experiments. This experimental method is unable to observe hydrogen atoms, and as a result, files with structures determined by x-ray crystallography do not include the atomic coordinates of hydrogen atoms.[13] NMR-determined structures, on the other hand, typically are able to determine the positions of hydrogen atoms, and thus include them in their data files.

The lack of hydrogens becomes a significant issue when creating models of desired macromolecules. The hydrogen atoms themselves are often key binding locations for custom

connectors. Further, the locations of these hydrogens can be necessary for the proper alignment of the connectors.

As such, if a file does not contain hydrogen atoms, they must be added in. The DNA duplex and RNA PDB-1DUL structures both required hydrogen addition. To that end, we have employed the open source molecular visualization software *UCSF ChimeraX.*[1] ChimeraX allows users to open any mmCIF and PDB file and visualize the resulting structure. The Add Hydrogen tool in ChimeraX can be used to protonate the structure.[14] Once complete, the file can be saved as an mmCIF.

It is important to note that Add Hydrogen determines hydrogen location via two methods: (1) steric only and (2) steric and hydrogen bond locations. Method (1), steric only, is based only on atom types and clash avoidance. Method (2) is the default and takes into account steric atom types and the location and angle of the hydrogen bonds. However, while Method (2) is the default, our exploration of the process shows that for macromolecules with unusual bonding patterns, when these unusual bonds are used to determine protonation, the hydrogen atoms are occasionally generated out of plane and twisted. Conversely, the steric method consistently returns hydrogen atoms in the locations predicted by structural convention. For this reason we have chosen to use the steric method in ChimeraX to add hydrogen atoms to our file. In addition to hydrogen atom management, ChimeraX can also be used to inspect and choose a subunit for conversion. When deciding what nucleic acid base on a chain constitutes a representative unit for modeling, reference to a molecular viewer of some kind is recommended.

Data Cleaning and Python Code

We have created a custom Python script for parsing and cleaning the atomic data, and we make it available as a Google Colab notebook. This script accelerates the process of parsing the mmCIF file to extract the atomic coordinates and export them as a CSV file. It further allows for export of specific chains or residues. To parse the mmCIF we made use of the Python library, *Biopython*, an open source set of tools for computational molecular biology.[5] The Parser's script made use of Biopython's Bio.PDB.MMCIFParser, Bio.PDB.MMCIF2Dict, and Bio.PDB.NeighborSearch modules to isolate, entities, models, chains, and residues available for modeling. The data was then scaled and arranged into lists for export as CSVs in the format described above in Table S1.

The benefit of the custom Parser is the high level of control a user has over how their mmCIF file is parsed, without requiring them to dig too deeply into the nuances of using Biopython themselves. Instead, a user merely runs the code in the browser and it prompts them through making decisions about what they would like to model. Since the custom Parser script has been coded to understand how to deal with files that have more than one type of model, chain, or entity, it enables the Parser to appropriately compute a large number of the molecular structural files available from the RCSB Protein Data Bank.

Taken altogether, the custom Parser and the MoleculeCrafter platform provide a highly accessible method for new users to create their own customized 3D printed, articulated models.

3D printing

The models shown in this paper were created using Shapeways (Eindhoven, NL), Xometry (North Bethesda, MD), two online vendors for on-demand manufacturing.[15] With their connectors and sockets, the unit parts are highly unusual and organic structures which can prove difficult to manufacture. However, these structures are easily printed using an isotropic process like Selective Laser Sintering (SLS). Our parts were made out of Nylon 12 and dyed a range of colors.

Access to MoleculeCrafter and other tools

(1) MoleculeCrafter Workspace: https://cad.onshape.com/documents/32717bfbc9f5e2ecec50465b/w/3d86996cbfc1ad808af54aa9/e/2672fefd7baba827b5c4f509

(2) MoleculeCrafter Workspace with models presented in this paper: https://cad.onshape.com/documents/d74a7ad9a22598f9e7764f8b/w/820dd43a092c4ab77328b76f/e/918e8638661d7272f968fb02?renderMode=0&uiState=6875229deeb4a14eac64e13a

(3) MoleculeCrafter Guided Parser: https://colab.research.google.com/drive/1l8c8AB8ofweAsAu5CAEN3BygvgQU1j-I?usp=drive_link#scrollTo=XSPfkbaD33x9

(4) The following instructional videos provide step-by-step tutorials illustrating entire model creation process:
   (a) MoleculeCrafter Tutorial Part 1 - Getting Molecular Data from a Database: https://youtu.be/L3CvjZhtVsA
   (b) MoleculeCrafter Tutorial Part 2 - Viewing the File in ChimeraX: https://youtu.be/wjtzOpmGY3Q
   (c) MoleculeCrafter Tutorial Part 3 - Parsing the Data: https://youtu.be/pLvbBbAczyI
   (d) MoleculeCrafter Tutorial Part 4 - Creating Molecules and Adding Connectors: https://youtu.be/ZQIg3FLuI0Y
   (e) MoleculeCrafter Tutorial Part 5 - Scaling, Aligning, and Stacking https://youtu.be/-2-jZup85Xw

Summary of stack (STL and STEP) files to create 1BNA, 139D, 1BWG, and 1DUL

This file contains step files which may be used to create 3D-printed models of the following DNA and RNA molecular structures: 1BNA, 139D, 1BWG, and 1DUL.

What follows here is a summary of the files and what they include.

1. 1BNA - Adenine (x10 bases)
2. 1BNA - Cytosine (x14 bases)
3. 1BNA - Guanine (x14 bases)
4. 1BNA - Thymine (x10 bases)

5. 1BWG Triplet - Adenine (x6 bases)
6. 1BWG Triplet - Cytosine (Hoogsteen) (x6 bases)
7. 1BWG Triplet - Cytosine (WCF) (x6 bases)
8. 1BWG Triplet - Guanine (x6 bases)
9. 1BWG Triplet - Thymine (Hoogsteen) (x6 bases)
10. 1BWG Triplet - Thymine (WCF) (x6 bases)

11. 1DUL - Adenine Combos (x13 bases of various types)
12. 1DUL - Cytosine Combos (x12 bases of various types)
13. 1DUL - Guanine Combos (x15 bases of various types)
14. 1DUL - Uracil Combos (x7 bases of various types)

15. 139D Quadruplet - Guanine (x16 bases)
16. 139D Quadruplet - Thymine (x12 bases)

17. 16x DNA Backbone (x16 backbones)
18. 139D - Backbone (x16 backbones)
19. 1DUL - Backbone (x30 backbones standard, x1 backbone special)

Stacks to make 1BNA print
• 1BNA - Adenine (x1 stack)
• 1BNA - Cytosine (x1 stack)
• 1BNA - Guanine (x1 stack)
• 1BNA - Thymine (x1 stack)
• 16x DNA Backbone (x2 stacks)* or (x4 stacks when making 1BNA & 1BWG)*

*Note, only x2 stack of backbones necessary to make 1BNA structure. However, there are more A, C, G, and T bases than necessary to make 1BNA. Print x4 backbone stacks to have enough backbones for all of the bases you print.

This was done because 1BNA and 1BWG use some of the same parts. It is often best to print both kits together at the same time. See below for more details.

Stacks to make 1BWG print
• 1BWG Triplet - Adenine (x1 stack)
• 1BWG Triplet - Cytosine (Hoogsteen) (x1 stack)
• 1BWG Triplet - Cytosine (WCF) (x1 stack)
• 1BWG Triplet - Guanine (x1 stack)
• 1BWG Triplet - Thymine (Hoogsteen) (x1 stack)
• 1BWG Triplet - Thymine (WCF) (x1 stack)
• 1BNA - Adenine (x1 stack)**
• 1BNA - Cytosine (x1 stack)**
• 1BNA - Guanine (x1 stack)**
• 1BNA - Thymine (x1 stack)**
• 16x DNA Backbone (x3 stacks)* or (x4 stacks when making 1BNA & 1BWG)*

**Note, if you are making 1BNA as well, you will not need to reprint another stack of 1BNA bases. There are enough bases in the 1BNA stacks to make the 1BNA structure and the 1BWG structure. However, you will need another backbone stack (x4 TOTAL for 1BNA and 1BWG structures).

Also, these triplet stacks come with extra triplet parts.

To see the actual parts used in the structures, their pattern of assembly, and the quantities used, look at the following pages.

Stacks to make 139D print

- 139D Quadruplet - Guanine (x1 stack)
- 139D Quadruplet - Thymine (x1 stack)
- 139D - Backbone (x2 stacks)***

Stacks to make 1DUL print

- 1DUL - Adenine Combos (x1 stack)
- 1DUL - Cytosine Combos (x1 stack)
- 1DUL - Guanine Combos (x1 stack)
- 1DUL - Uracil Combos (x1 stack)
- 1DUL - Backbone (x2 stacks)***

***Will have extra backbones when printed.

Assembly pattern for 1BNA

C (x8), G (x8), A (x4), T (x4), Backbone, DNA (x24)

5’	3’
C	G
G	C
C	G
G	C
A	T
A	T
T	A
T	A
C	G
G	C
C	G
G	C
3’	5’

Assembly pattern for 1BWG

C (x4), C triplet (WCF) (x3), C triplet (Hoogsteen) (x3),
G (x4), G triplet (x3),
A (x4), A triplet (x3),
T (x4), T triplet (WCF) (x3), T triplet (Hoogsteen) (x3)
Backbone, DNA (x34)

5’	3’	
T	A	
A	T	
C	G	
G	C	3’
T (triplet, WCF)	A (triplet)	T (triplet, Hoogsteen)
C (triplet, WCF)	G (triplet)	C (triplet, Hoogsteen)
T (triplet, WCF)	A (triplet)	T (triplet, Hoogsteen)
C (triplet, WCF)	G (triplet)	C (triplet, Hoogsteen)
T (triplet, WCF)	A (triplet)	T (triplet, Hoogsteen)
C (triplet, WCF)	G (triplet)	C (triplet, Hoogsteen)
A	T	5’
G	C	
T	A	
C	G	
3’	5’	

Assembly pattern for 139D

T quadruplet (x12), G quadruplet (x16), 139D - Backbone (x28)

5’	5’	5’	5’
T (No H bonding)	T (No H bonding)	T (No H bonding)	T (No H bonding)
T	T	T	T
G	G	G	G
G	G	G	G
G	G	G	G
G	G	G	G
T	T	T	T
3’	3’	3’	3’

*No Hydrogen bonding will occur between Thymine bases labeled (No H bonding). These only have bonding between bases and backbones and backbones and backbones.

Assembly pattern for 1DUL

C (x12), G (x15), A (x13), U (x7), 1DUL - Backbone (x46 plus x1 special)

5’	Bond Type	3’
G	WCF - WCF	C
C	WCF - WCF	G
U	WCF - WCF	A
C	WCF - WCF	G
U	WCF - WCF	A
G	WCF - WCF	C
U	WCF - WCF (Wobble)	G
U	WCF - WCF (Wobble)	G
U	WCF - WCF	A
A		
		A

C		
C		
A		
G	WCF - WCF	C
G	WCF - WCF	C
U	WCF - WCF (Wobble)	G
C (Has special backbone)	Sugar - Hoogsteen	A
A	Hoogsteen - WCF	C
G	Hoogsteen - Sugar	G
G	WCF - WCF (Wobble, Cis)	A
U	WCF - WCF	A
C	WCF - WCF	G
C	WCF - WCF	G
G	Sugar - Hoogsteen	A
A		A

* Note: Technically the 1DUL structure is one long strand that loops back on itself. However, the two Adenines at the bottom do not always connect to form one nice strand. This is a minor limitation of the current model.

REFERENCES

1. Meng, E. C. *et al.* UCSF ChimeraX: Tools for structure building and analysis. *Protein Sci. Publ. Protein Soc.* **32**, e4792 (2023).
2. Lu, X.-J., Bussemaker, H. J. & Olson, W. K. DSSR: an integrated software tool for dissecting the spatial structure of RNA. *Nucleic Acids Res.* **43**, e142 (2015).
3. Lu, X.-J. & Olson, W. K. 3DNA: a versatile, integrated software system for the analysis, rebuilding and visualization of three-dimensional nucleic-acid structures. *Nat. Protoc.* **3**, 1213–1227 (2008).
4. Drew, H. R. *et al.* Structure of a B-DNA dodecamer: conformation and dynamics. *Proc. Natl. Acad. Sci. U. S. A.* **78**, 2179–2183 (1981).
5. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinforma. Oxf. Engl.* **25**, 1422–1423 (2009).
6. Goodwin-Schoen, C. & Taylor, R. E. Modular, articulated models of DNA and peptide nucleic acids for nanotechnology education. *The Biophysicist* **?**, ? (2023).
7. Onshape, a PTC Business. Why Onshape? *Why Onshape?* https://www.onshape.com/en/why-onshape.
8. Goodwin-Schoen, C. M. & Taylor, R. E. Modular, Articulated Models of DNA and Peptide Nucleic Acids for Nanotechnology Education. *The Biophysicist* **4**, 1–10 (2023).
9. Mantina, M., Chamberlin, A. C., Valero, R., Cramer, C. J. & Truhlar, D. G. Consistent van der Waals Radii for the Whole Main Group. *J. Phys. Chem. A* **113**, 5806–5812 (2009).
10. RCSB Protein Data Bank. RCSB PDB: Homepage. https://www.rcsb.org/.
11. RCSB PDB-101. PDB101: Learn: Guide to Understanding PDB Data: Beginner's Guide to PDBx/mmCIF. *RCSB: PDB-101* https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/beginner%E2%80%99s-guide

-to-pdbx-mmcif.

12. Lawson, C. L., Berman, H. M., Chen, L., Vallat, B. & Zirbel, C. L. The Nucleic Acid Knowledgebase: a new portal for 3D structural information about nucleic acids. *Nucleic Acids Res.* **52**, D245–D254 (2024).

13. RCSB PDB-101. PDB101: Learn: Guide to Understanding PDB Data: Missing Coordinates. *RCSB: PDB-101* https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/missing-coordinates.

14. UCSF ChimeraX. Tool: Add Hydrogens. https://www.cgl.ucsf.edu/chimerax/docs/user/tools/addhydrogens.html.

15. Xometry. Introduction to Xometry. *Xometry's Manufacturing Community* https://community.xometry.com/kb/articles/496-introduction-to-xometry (2024).