

HELM Antibody Editor

User Guide V2

© 2017 Stefan Klostermann, pREDi Roche Innovation Center Munich

Table of Contents

I Overview	4
1 General.....	4
2 HELM Antibody Editor appearance.....	4
Main form	4
Sequence editor	5
Domain detection window	6
Antibody editing window	7
Legend for assembled antibodies	7
The HELM Editor	8
II HowTo	10
1 Start HELM Antibody Editor.....	10
2 Load sequences.....	10
3 Find domains.....	10
4 Analyse domains.....	16
5 Edit antibodies and single domains.....	17
Sortase Coupling	17
ADC designer	17
MonomerDesigner	18
Domain Protease Reaction	19
Biotinylation	20
Connect domains manually.....	21
Edit domains in the HELM Editor	22
6 Save final antibody.....	33
7 Load registered antibody.....	33
III Antibody Editor Settings	34
1 Domain Detection Settings.....	35
2 Libraries.....	37
IV Antibody Editing Window	39
1 Cys-Cys bond creation.....	39
2 Canvas options.....	39
V Libraries	41
1 Domain library.....	41
2 Mutation library.....	42
3 Autoconnector rule set.....	43
4 Protease library.....	44

VI Contact

46

1 Overview

1.1 General

The *HELM Antibody Editor* V2.0 is a separate application that employs HELM technology to handle complex antibodies for their analysis, assembly, visualization, manipulation and registration.

Beginning from the raw sequence, the entered peptide chains are chopped into domains using the Basic Local Alignment Search Tool (BLAST) versus the domain library. BLAST finds regions of local similarity between sequences of the domains stored in the library and the entered raw sequence. The resulting domain hits are filtered for their %identity and %coverage, sorted and finally assembled to an antibody or antibody-like molecule. You can chemically modify domains of this antibody, e.g. to add payloads, based on *HELM Editor* V1.2. Please note that the test sequences provided in the folder "sample" are artificial sequences designed just to exemplify the different features.

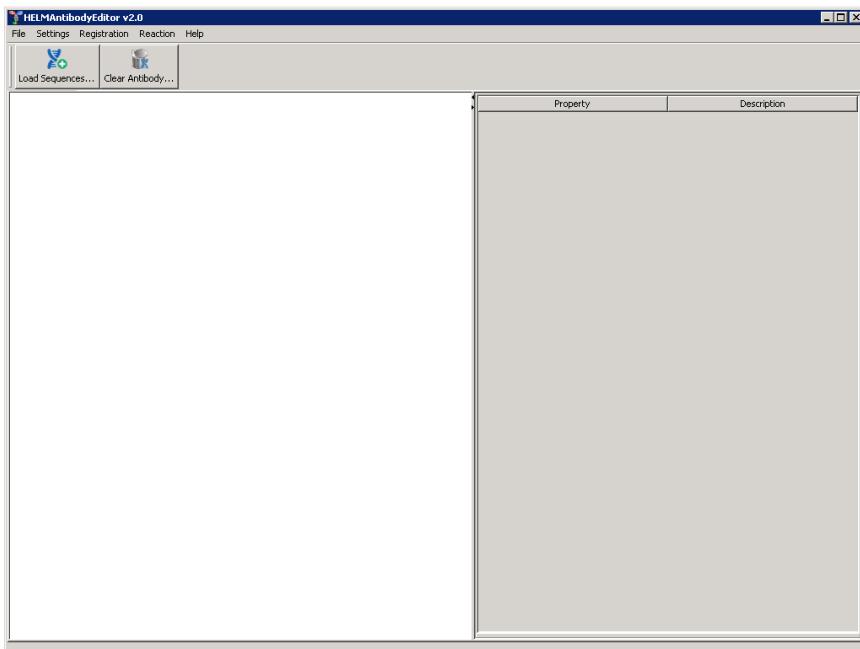
1.2 HELM Antibody Editor appearance

The *HELM Antibody Editor* includes several operation windows with different functionality and various detail boxes containing specific information.

Their appearance and basic functionality are introduced in the following, while the full functionality is explained in chapter II.

1.2.1 Main form

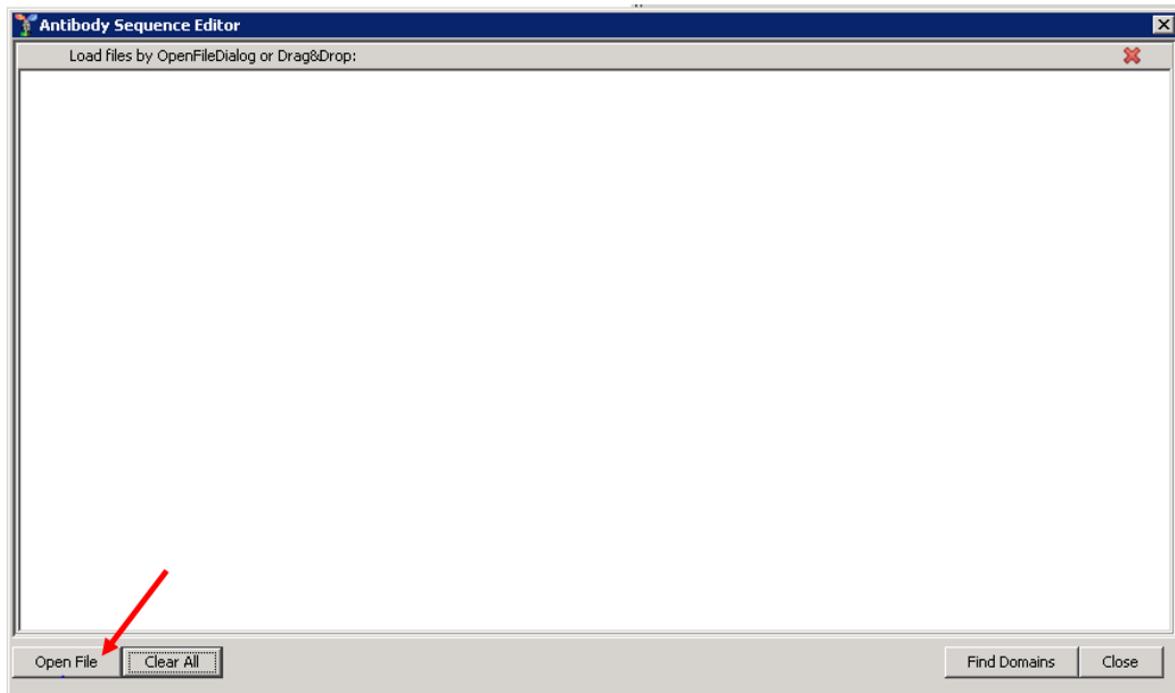
After starting the *HELM Antibody Editor*, the main form will show up.



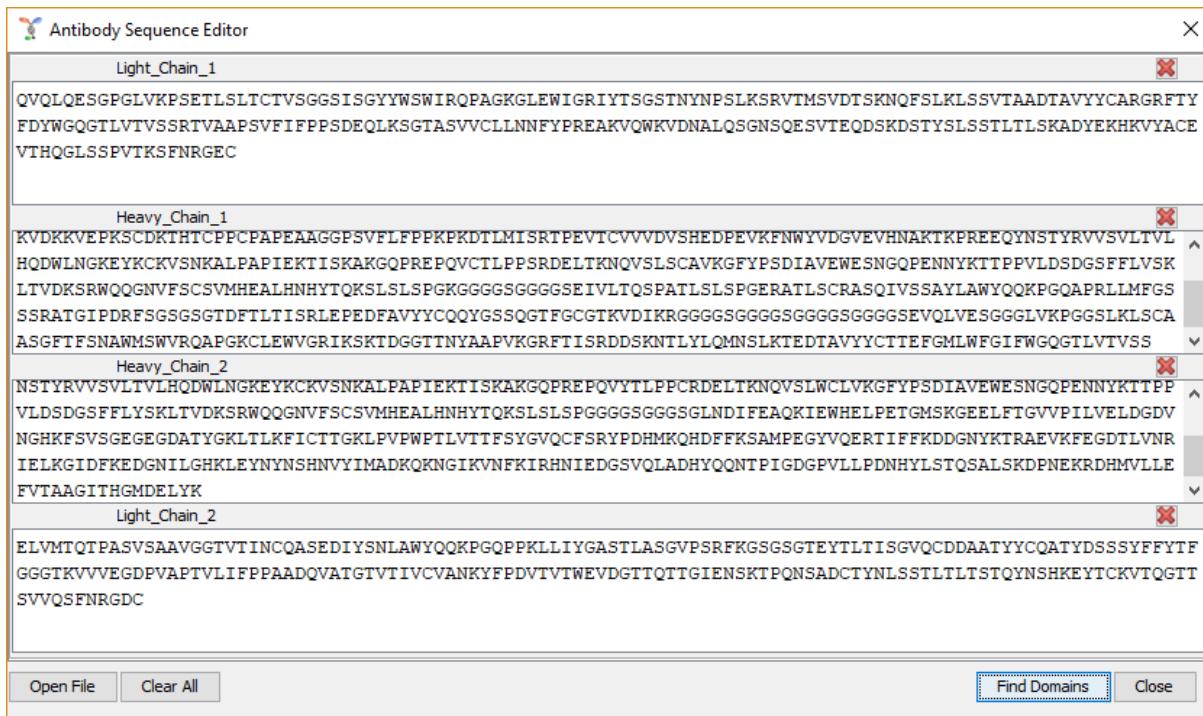
The toolbar contains the [Load Sequences...] tab, with which you can upload your raw sequence.

1.2.2 Sequence editor

Clicking on the [Load sequences...] tab opens the "Antibody Sequence Editor".



After loading your raw sequences, e.g. "sample\tri-specific test antibody.fa" from your HELMAntibodyEditor folder, with the [Open File] button or by dragging the FASTA formatted file(s) into the window, they are shown in the window.

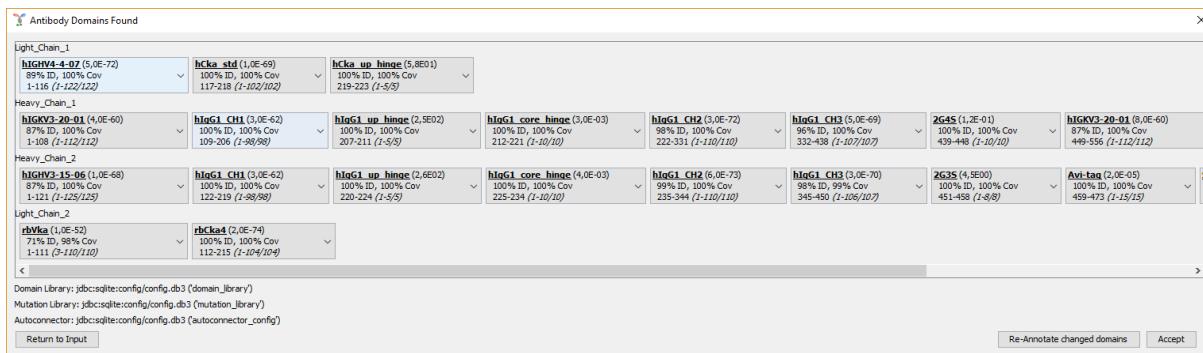


Please go to [Chapter 2.2 \(Load sequences\)](#) for further information concerning the sequence loading process.

Besides the [Open File], [Clear All] and [Close] buttons, the "Antibody Sequence Editor" includes the [Find Domains] button, which leads to the domain detection window.

1.2.3 Domain detection window

Clicking on the [Find Domains] button in the sequence editor opens the "Antibody Domains Found" window.



The most significant domain hits are shown in boxes, including different parameters:

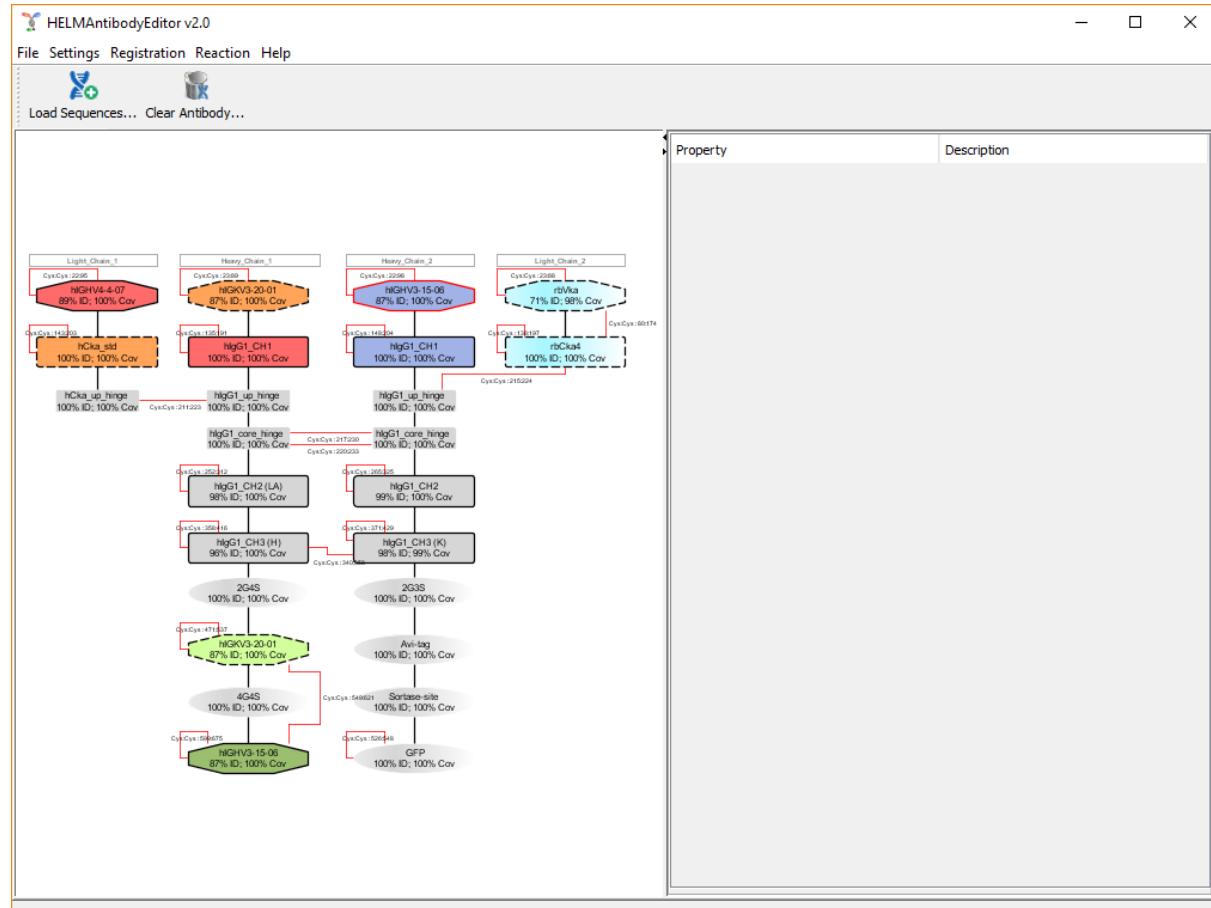
hIgG1_CH1 (3,0E-62)
100% ID, 100% Cov
109-206 (1-98/98)

Each domain found through BLAST is named with the domain short name delivered by the domain library, displayed in bold type. For more information about the other variables see Chapter 3.2 (Domain Detection Settings). More details about the opportunities of the "Antibody Domains Found" window are delivered in Chapter 2.3 (Find domains).

In order to assemble the found domains to the appropriate antibody, you have to click on [Accept] in the right bottom corner of the domain detection window.

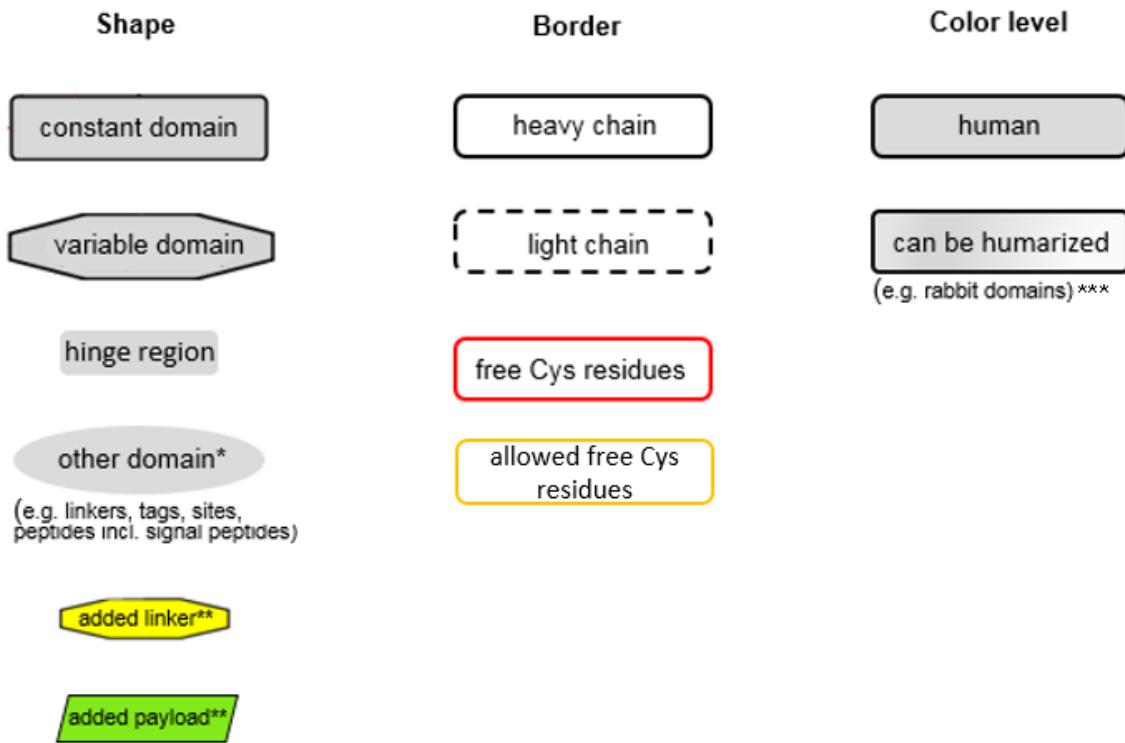
1.2.4 Antibody editing window

Clicking on [Accept] in the domain detection window opens the antibody editing window with your assembled antibody. This example antibody is assembled automatically, thanks to the auto-connector ruleset



1.2.5 Legend for assembled antibodies

As you can see above, the domains differ in their border type and color, shape and color level. This is intended to facilitate a quick classification of each domain to its domain type.



* not present in the sample antibody

** edited chemical modifications added with the HELM Editor

*** note the gradient

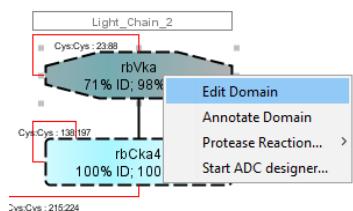
Red lines indicate interdomain and intradomain Cys-Cys bridges. The application tries to automatically establish as many of these bridges as possible, remaining ones have to be done manually.

The upper example is a fake antibody with a rabbit light chain to the left. Note for demonstration purposes in addition to the intradomain Cys-Cys bridge both "rbVka" and "rbCka4" domains contain an additional Cys that form an additional bridge and "rbVka" contains even another (4th) Cys (located in the CDR3).

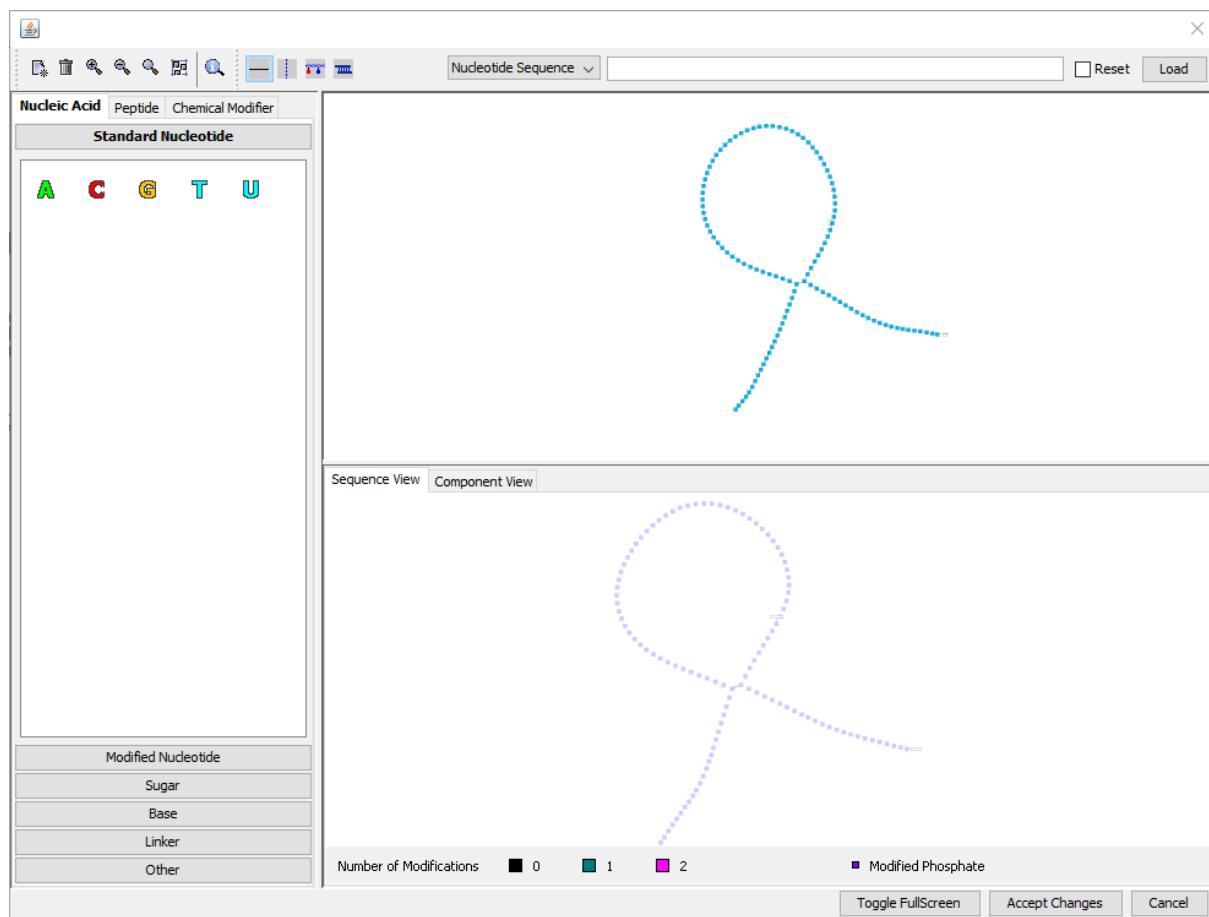
Paired Fv regions are marked by a common background color. Orange lines indicate that the domain has allowed free cysteins.

1.2.6 The HELM Editor

You can edit domains i.e. modify them chemically. These chemical modifications are enabled by the *HELM Editor* V1.2, which opens when you right-click a certain domain and choose [Edit Domain].



Note that calling the HELM Editor the 1st time installs a folder for the monomer library in C:\Users \<user>\.helm if not already done.



See [Chapter 2.5.7 Edit domains in the HELM Editor](#) for further information about the *HELM Editor* and certain steps of domain editing.

2 HowTo

As a practical illustration of the new *HELM Antibody Editor* functionality the following section uses a provided sample antibody and showcases the steps and options in more detail.

2.1 Start HELM Antibody Editor

You start the HELM Antibody Editor by running the HELMAntibodyEditor.jar file. The main form opens as described under Main form. If you can't run the .jar file, you most likely need to install the Java Runtime Environment. You can download it free at <http://www.java.com/en/download/>.

2.2 Load sequences

To load your raw sequence click on [Load Sequences...] in the toolbar of the main form. The Antibody Sequence Editor opens as explained in [Chapter 1.2.2 \(Sequence editor\)](#). Select the [Open File] button and open "sample\tri-specific test antibody.fa" from your HELMAntibodyEditor folder. Valid file formats are (multiple) FASTA or Genbank format. Alternatively you can use the drag&drop function.

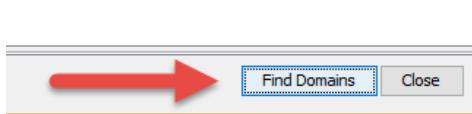
Both ways you will result in:



Your raw sequence has now been uploaded to the HELM Antibody Editor. To open a different antibody, first select [Clear all]. Otherwise, the sequences will be added to the existing sequences.

2.3 Find domains

Click [Find Domains] in the sequence editor to match your entered peptide chain with the domain library.



Pursuant to your domain detection settings the corresponding domains are shown in the "Antibody Domain Found" window:

Antibody Domains Found

Light_Chain_1	hck4_std (1.0E-69)	hck4_up_hinge (5.8E01)
hGK4-4-02 (4.0E-72) 89% ID, 100% Cov 1-116 (1-122/122)	100% ID, 100% Cov 117-218 (1-102/102)	100% ID, 100% Cov 219-223 (1-5/5)
Heavy_Chain_1		
hGKV3-20-01 (4.0E-60) 87% ID, 100% Cov 1-108 (1-112/112)	hlg1_C1H1 (3.0E-62) 100% ID, 100% Cov 109-206 (1-98/98)	hlg1_up_hinge (2.5E02) 100% ID, 100% Cov 207-211 (1-5/5)
hlg1_C1H2 (3.0E-72) 100% ID, 100% Cov 212-221 (1-10/10)	hlg1_core_hinge (3.0E-03) 100% ID, 100% Cov 222-331 (1-10/10)	hlg1_C1H3 (5.0E-69) 98% ID, 100% Cov 332-438 (1-10/10)
hlg1_C2 (1.2E-01) 100% ID, 100% Cov 439-448 (1-10/10)	2G4S (4.5E00) 100% ID, 100% Cov 451-459 (1-9/9)	hGKV3-20-01 (8.0E-60) 87% ID, 100% Cov 449-556 (1-112/112)
Heavy_Chain_2		
hGKV3-15-06 (1.0E-68) 87% ID, 100% Cov 1-121 (1-125/125)	hlg1_C1H1 (3.0E-62) 100% ID, 100% Cov 122-219 (1-98/98)	hlg1_up_hinge (2.6E02) 100% ID, 100% Cov 220-224 (1-5/5)
hlg1_C1H2 (6.0E-73) 100% ID, 100% Cov 225-234 (1-10/10)	hlg1_core_hinge (4.0E-03) 100% ID, 100% Cov 235-344 (1-10/10)	hlg1_C1H3 (3.0E-70) 98% ID, 99% Cov 345-450 (1-10/10)
2G3S (4.5E00) 100% ID, 100% Cov 451-459 (1-9/9)	Avi-tag (2.0E-05) 100% ID, 100% Cov 459-473 (1-15/15)	
Light_Chain_2		
rck4 (1.0E-52) 71% ID, 98% Cov 1-111 (3-110/110)	rck4 (2.0E-74) 100% ID, 100% Cov 112-215 (1-104/104)	

Domain Library: jdbcsqlite:config/config.db3 (domain_library)
Mutation Library: jdbcsqlite:config/config.db3 (mutation_library)
Autoconnector: jdbcsqlite:config/config.db3 ('autoconnector_config')

Return to Input Re-Annotate changed domains Accept

To see the domain sequence, move the mouse over domains:

Antibody Domains Found

Light_Chain_1	hck4_std (1.0E-69)	hck4_up_hinge (5.8E01)
hGK4-4-02 (4.0E-72) 89% ID, 100% Cov 1-116 (1-122/122)	100% ID, 100% Cov 117-218 (1-102/102)	100% ID, 100% Cov 219-223 (1-5/5)
Heavy_Chain_1		
hGKV3-20-01 (4.0E-60) 87% ID, 100% Cov 1-108 (1-112/112)	hlg1_C1H1 (3.0E-62) 100% ID, 100% Cov 109-206 (1-98/98)	hlg1_up_hinge (2.5E02) 100% ID, 100% Cov 207-211 (1-5/5)
hlg1_core_hinge (3.0E-03) 100% ID, 100% Cov 212-221 (1-10/10)	hlg1_C1H2 (3.0E-72) 98% ID, 100% Cov 222-331 (1-10/10)	hlg1_C1H3 (5.0E-69) 98% ID, 100% Cov 332-438 (1-10/10)
2G4S (1.2E-01) 100% ID, 100% Cov 439-448 (1-10/10)	hGKV3-20-01 (8.0E-60) 87% ID, 100% Cov 449-556 (1-112/112)	4G4S (4.0E-09) 100% ID, 100% Cov 538-577 (1-20/20)
Heavy_Chain_2		
hGKV3-15-06 (1.0E-68) 87% ID, 100% Cov 1-121 (1-125/125)	hlg1_C1H1 (3.0E-62) 100% ID, 100% Cov 122-219 (1-98/98)	hlg1_up_hinge (2.6E02) 100% ID, 100% Cov 220-224 (1-5/5)
hlg1_core_hinge (4.0E-03) 100% ID, 100% Cov 225-344 (1-10/10)	hlg1_C1H2 (6.0E-73) 99% ID, 100% Cov 345-450 (1-10/10)	hlg1_C1H3 (3.0E-70) 98% ID, 99% Cov 451-459 (1-9/9)
2G3S (4.5E00) 100% ID, 100% Cov 451-459 (1-9/9)	Avi-tag (2.0E-05) 100% ID, 100% Cov 459-473 (1-15/15)	GFP (0.0E00) 100% ID, 100% Cov 479-498 (1-238/238)
Light_Chain_2		
rck4 (1.0E-52) 71% ID, 98% Cov 1-111 (3-110/110)	rck4 (2.0E-74) 100% ID, 100% Cov 112-215 (1-104/104)	

Domain Library: jdbcsqlite:config/config.db3 (domain_library)
Mutation Library: jdbcsqlite:config/config.db3 (mutation_library)
Autoconnector: jdbcsqlite:config/config.db3 ('autoconnector_config')

Return to Input Re-Annotate changed domains Accept

If you click a domain, another window opens, firstly showing further BLAST hits from the same regions. These are listed below according to their sorting order, which you can determine in the [Domain Detection Settings \(Chapter 3.1\)](#) with their colors indicating their sorting blocks :

The screenshot shows a list of domains on the left and a grid of domain pairs on the right. The domains listed are:

- hIGHV3-73-02** (9,0E-56)
76% ID, 100% Cov
1-121 (1-125/125)
- hIGHV3-72-01** (3,0E-56)
74% ID, 100% Cov
1-121 (1-125/125)
- hIGHV3-49-04** (2,0E-54)
74% ID, 100% Cov
1-121 (1-125/125)
- hIGHV3-49-05** (2,0E-54)
74% ID, 100% Cov
1-121 (1-125/125)
- hIGHV3-49-02** (5,0E-54)
74% ID, 100% Cov
1-121 (1-125/125)
- hIGHV3-49-03** (8,0E-54)
74% ID, 100% Cov
1-121 (1-125/125)
- hIGHV3-74-02** (5,0E-53)
72% ID, 100% Cov
1-121 (1-123/123)
1-121 (1-125/125) (highlighted by a red arrow)

The grid of domain pairs includes:

hCka_std (1,0E-69) 100% ID, 100% Cov 117-218 (1-102/102)	hCka_up_hinge (5,8E01) 100% ID, 100% Cov 219-223 (1-5/5)		
hIgG1_CH1 (3,0E-62) 100% ID, 100% Cov 109-206 (1-98/98)	hIgG1_up_hinge (2,5E02) 100% ID, 100% Cov 207-211 (1-5/5)	hIgG1_core_hinge (3,0E-03) 100% ID, 100% Cov 212-221 (1-10/10)	hIgG1 98% ID 222-33
hIgG1_CH1 (3,0E-62) 100% ID, 100% Cov 122-219 (1-98/98)	hIgG1_up_hinge (2,6E02) 100% ID, 100% Cov 220-224 (1-5/5)	hIgG1_core_hinge (4,0E-03) 100% ID, 100% Cov 225-234 (1-10/10)	hIgG1 99% ID 235-34

Below the grid, there are links to Domain Library, Mutation Library, and Autoconnector, and a "Return to Input" button.

Any of those domains can be selected to manually override the present top hit.
Additionally, following options are offered:

- **Assign to the left/ right:** assign a complete domain to the neighboring left/right one. Used if a (smaller) domain is obsolete and should be part of another neighboring one.
- **Restrict left/ right border to the neighboring one:** adjust left/right domain boundary to the border of the neighboring one such that the overlap is solved. Used in case of overlapping BLAST hits.
- **Edit boundaries:** manually edit the boundaries of a domain. Used to have full control over the domain boundaries.
- **None:** delete domain assignment in case the annotation is wrong. This domain can later be manually annotated in the Antibody Editing Window.

By clicking [Re-annotate modified domains] these domains are re-BLASTed and re-annotated.

Antibody Domains Found

Light_Chain_1	hIGHV4-4-07 (5,0E-72) 89% ID, 100% Cov 1-116 (1-122/122)	hCka_std (1,0E-69) 100% ID, 100% Cov 117-218 (1-102/102)	hCka_up_hinge (5,8E01) 100% ID, 100% Cov 219-223 (1-5/5)	
Heavy_Chain_1	hIGKV3-20-01 (4,0E-60) 87% ID, 100% Cov 1-108 (1-112/112)	hIgG1_CH1 (3,0E-62) 100% ID, 100% Cov 109-206 (1-98/98)	hIgG1_up_hinge (2,5E02) 100% ID, 100% Cov 207-211 (1-5/5)	hIgG1_core_hinge 100% ID, 100% Co 212-221 (1-10/10)
Heavy_Chain_2	hIGHV3-15-06 (1,0E-68) 87% ID, 100% Cov 1-121 (1-125/125)	hIgG1_CH1 (3,0E-62) 100% ID, 100% Cov 122-219 (1-98/98)	hIgG1_up_hinge (2,6E02) 100% ID, 100% Cov 220-224 (1-5/5)	hIgG1_core_hinge 100% ID, 100% Co 225-234 (1-10/10)
Light_Chain_2	rbVka (1,0E-52) 71% ID, 98% Cov 1-111 (3-110/110)	rbCka4 (2,0E-74) 100% ID, 100% Cov 112-215 (1-104/104)		

Domain Library: jdbc:sqlite:config/config.db3 ('domain_library')
Mutation Library: jdbc:sqlite:config/config.db3 ('mutation_library')
Autoconnector: jdbc:sqlite:config/config.db3 ('autoconnector_config')

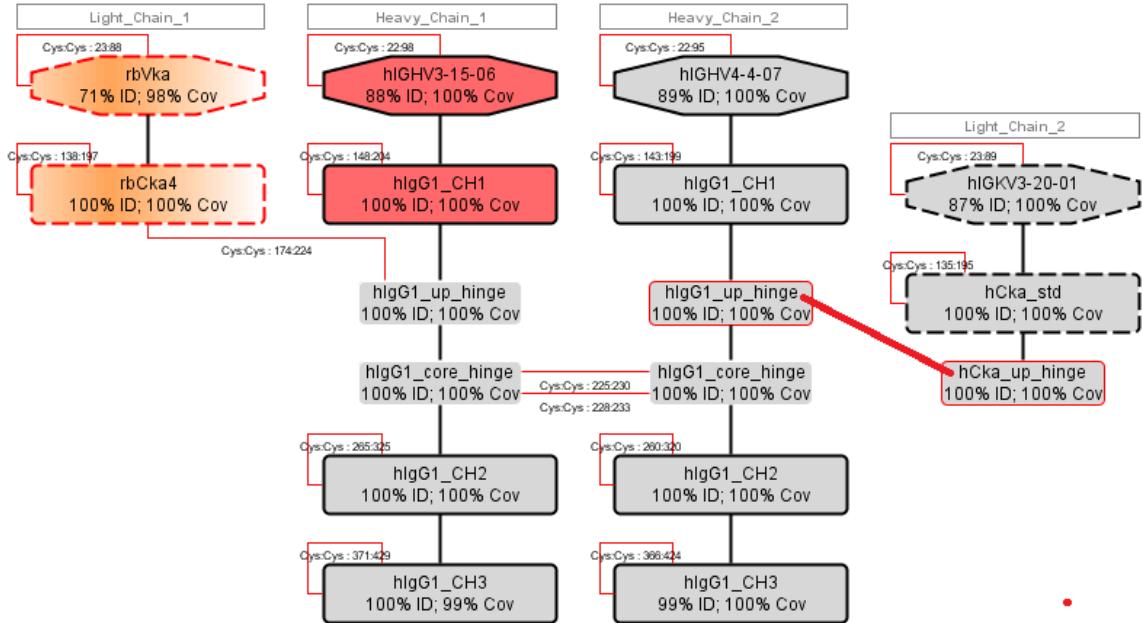
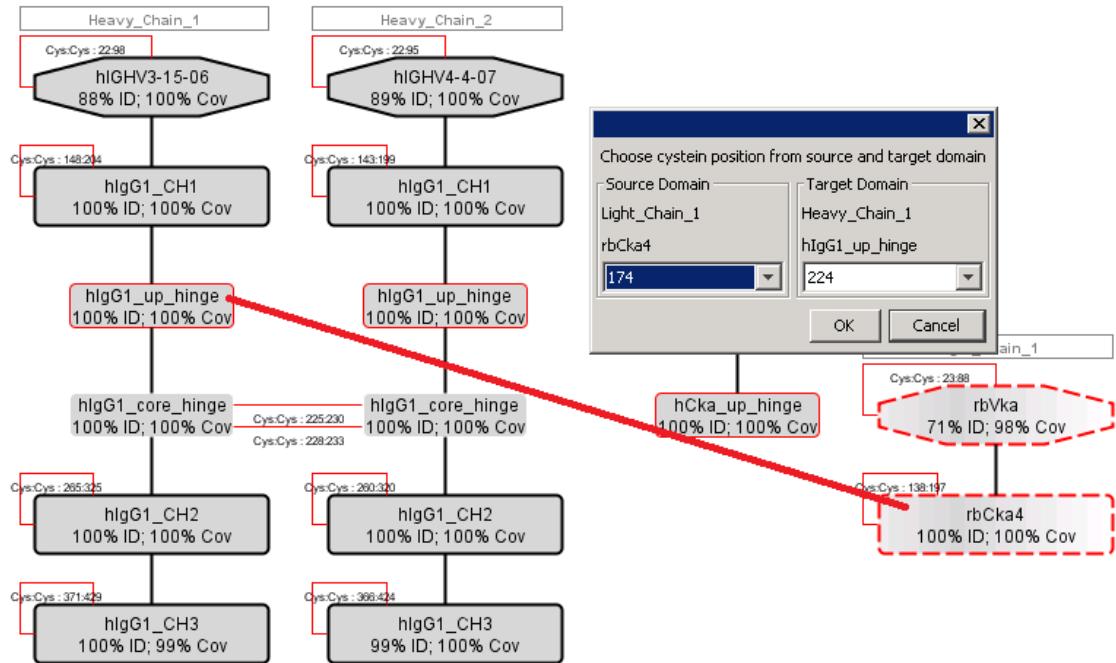
The button [Return to Input] returns to the Sequence Loader. [Accept] transfers the domains to the Antibody Editing Window, where the antibody is visualized:

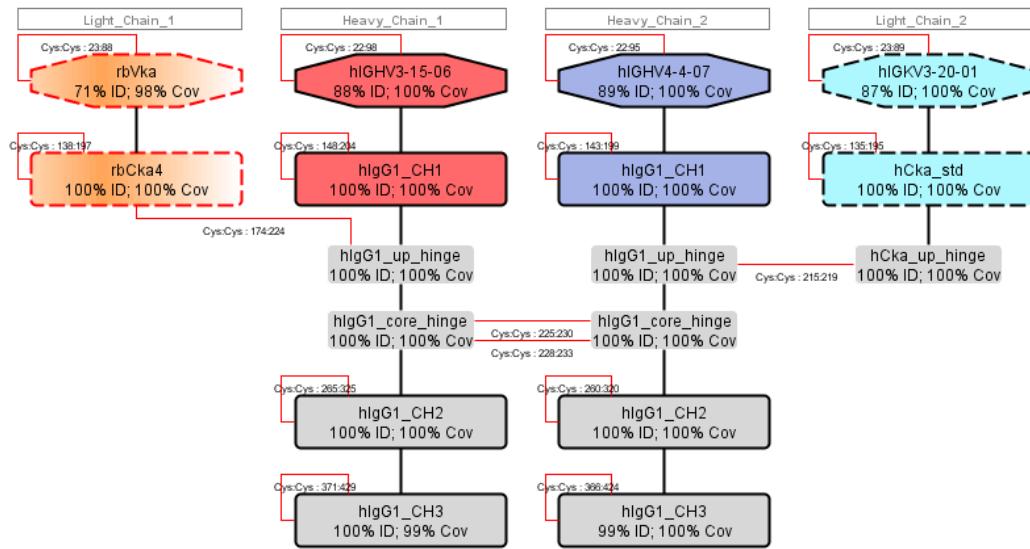


Now you can accurately analyze and edit the domains. The antibody above was automatically connected, due to the autoconnector ruleset.

If your antibody was NOT automatically connected, you can now connect the chains, see [chapter 2.5.3 Connect domains manually](#). In the example below ("sample\mixing test antibody.fa" from your HELMAntibodyEditor folder), the antibody was not automatically connected, because the two different light chains cannot be unequivocally paired to the two different heavy chains.

Manually connecting the **rbCka4** of the **Light_Chain_1** with the **hlgG1_up_hinge** of the **Heavy_Chain_1** and the **hCka_up_hinge** of the **Light_Chain_2** with the **hlgG1_up_hinge** of the **Heavy_Chain_2** results in a full antibody.





2.4 Analyse domains

Click on a certain domain to access additional information. Its details box will open.

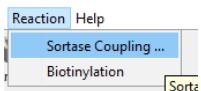
Property	Description
> CHAIN	Light_Chain_2
NAME	
SEQUENCE LENGTH	215
SEQUENCE	ELVMTQTPASVAAVGGTVTINCQASEDLYSNL...
ORIGINAL SEQUENCE	ELVMTQTPASVAAVGGTVTINCQASEDLYSNL...
> DOMAIN	
USER LABEL	rbVka
USER COMMENT	71% ID; 98% Cov
START POS ON CHAIN	1
END POS ON CHAIN	111
LENGTH	111
START POS ON TEMPLATE	3
END POS ON TEMPLATE	110
IDENTITY TO LIB. DOH.	71%
COVERAGE OF LIB. DOH.	98%
LEADING SEQUENCE	SEQUENCE (DE FACTO)
	ELVMTQTPASVAAVGGTVTINCQASEDLYSNL...
TRAILING SEQUENCE	
	2
AUTOMORPHOUS	NO
> ANNOTATIONS	
rbV-L-L	
scFv_VL-VH	
up-hinges_1x	
Std-Ab_CLx_1x	
> MUTATIONS	
> UNKNOWN MUTATIONS	
> CYS BRIDGES	
CYS POS	[23, 80, 88]
CYS BRIDGES	[rbVka:23 <-> rbVka:88, rbVka:80 <-> rbV...
FREE CYS	0
> OTHER BRIDGES	
GENERAL	0
> RECOGNIZED DOMAIN FROM LIBRARY	
NAME IN LIB	iKAPPA-CONS_RABBIT
SHORT NAME	rbVka
SPECIES	Rabbit (<i>Oryctolagus cuniculus</i>)
HUMANNESS	Humanizable
CHAIN	Kappa
DOMAIN TYPE	VARIABLE
SEQUENCE LENGTH IN LIB	110
CYS COUNT	4
FREE_CYS_ALLOWED	0
PATTERN	[1-3]
COMMENT	
SEQUENCE (TEMPLATE)	DPVLTQTPSPVSAANGTVTISCAQSQSYNN...
DOMAIN USAGE	CORE

This box contains all data of the full chain (>CHAIN), the domain found and analyzed as entered (>DOMAIN), recognized formats (>ANNOTATIONS) and known and unknown mutations (>MUTATIONS, >UNKNOWN MUTATIONS). Furthermore information about all, connected and free, Cysteins (>CYS BRIDGES) and every other bond e.g. to chemical linkers (>OTHER BRIDGES) is listed. Data of the the domain from domain library, which has been used to annotate the given domain, is also shown (>RECOGNIZED DOMAIN FROM LIBRARY).

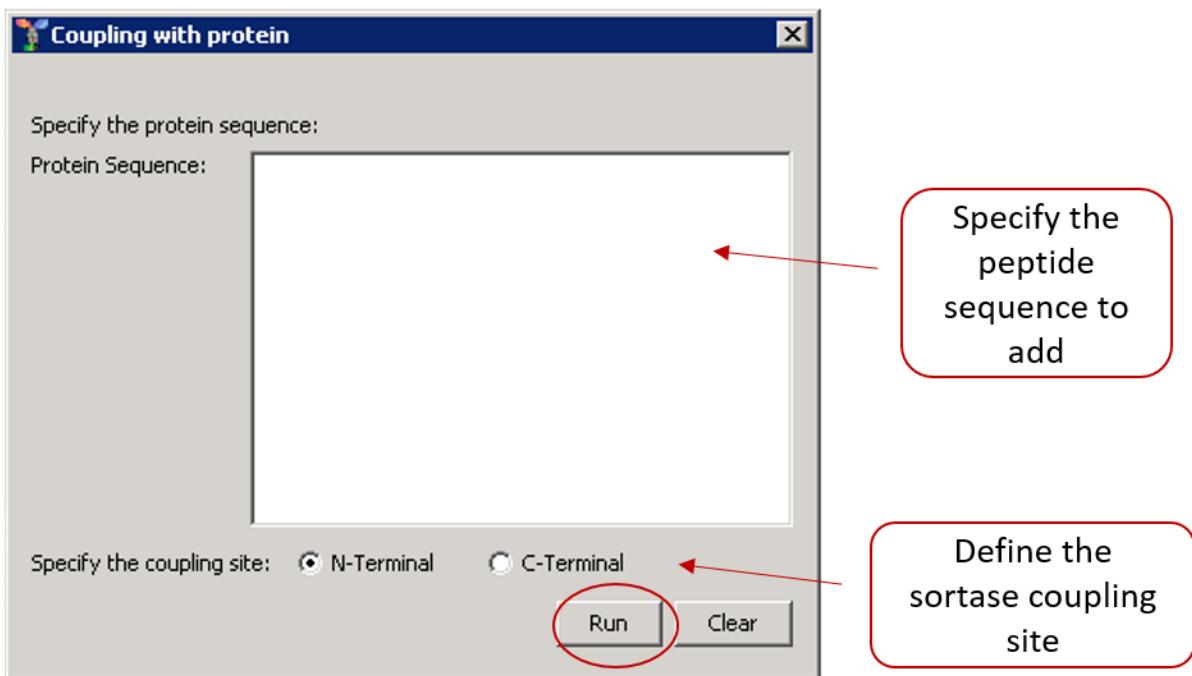
2.5 Edit antibodies and single domains

2.5.1 Sortase Coupling

You can couple peptide sequence to your antibody. By clicking on the [Sortase Coupling] button in the [Reaction] menu opens the "Coupling with protein" window.

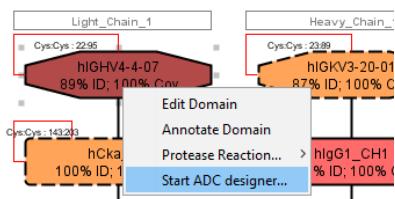


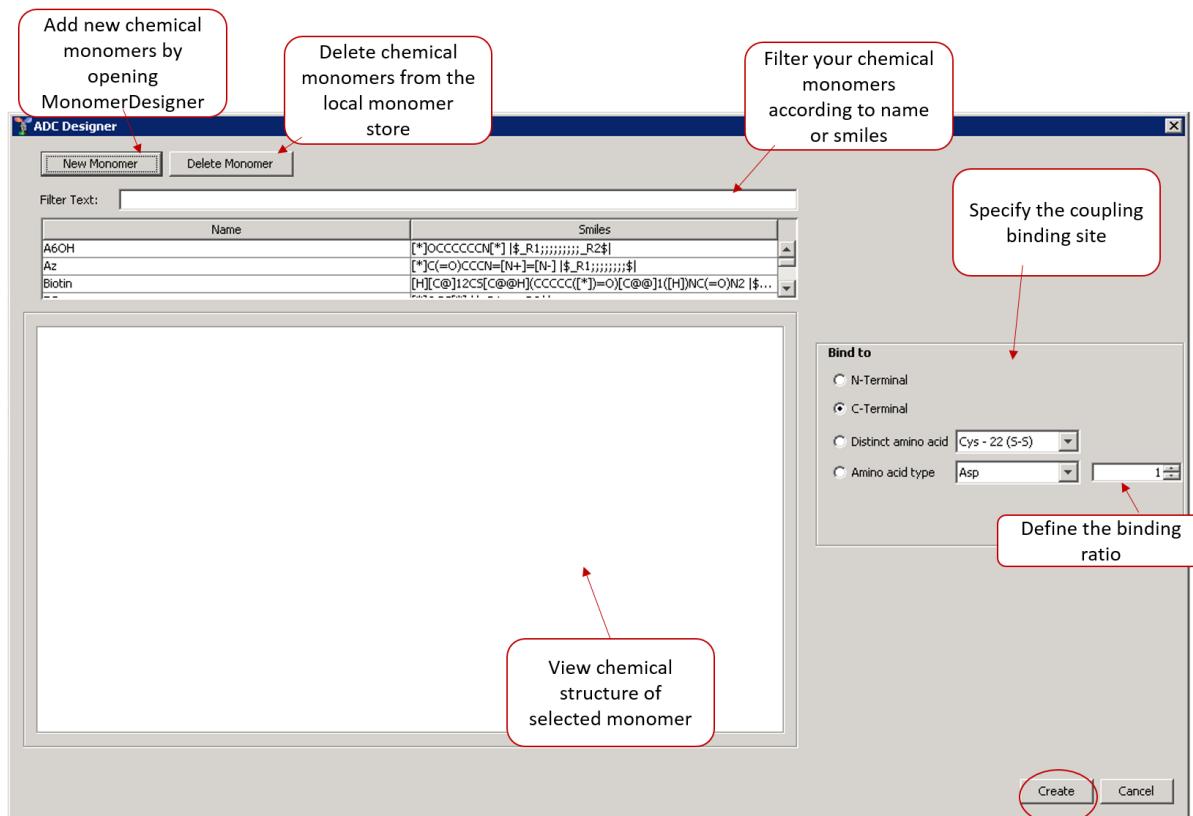
The given peptide sequence will be added n- or c-terminal to all cutted sortase-sites of the loaded antibody.



2.5.2 ADC designer

You can edit domains i.e. modify them chemically. These chemical molecules are attached to the selected domain. Right-clicking a certain domain and choosing [Start ADCDesigner] opens "ADC designer" window.



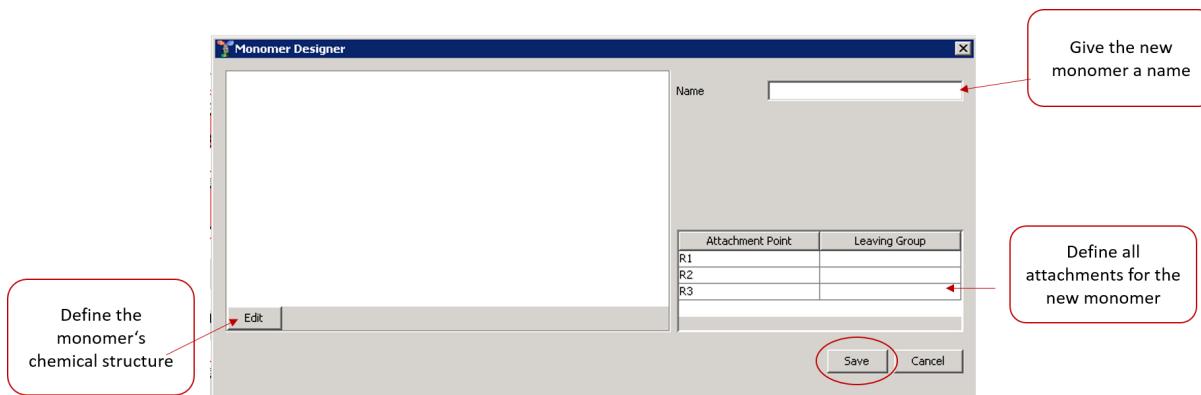


The ADC designer allows you to add and/or delete chemical monomers from your local monomer store. Also you can filter the monomers according to the name or the SMILES. Selecting one single chemical monomer shows its chemical structure. The selected monomer will be attached to the domain at your specified coupling binding site. Here, there are four different options. The monomer can be attached N- or C-terminally to the specified domain. You might also choose a specific amino acid or an amino acid type with a binding ratio.

2.5.3 MonomerDesigner

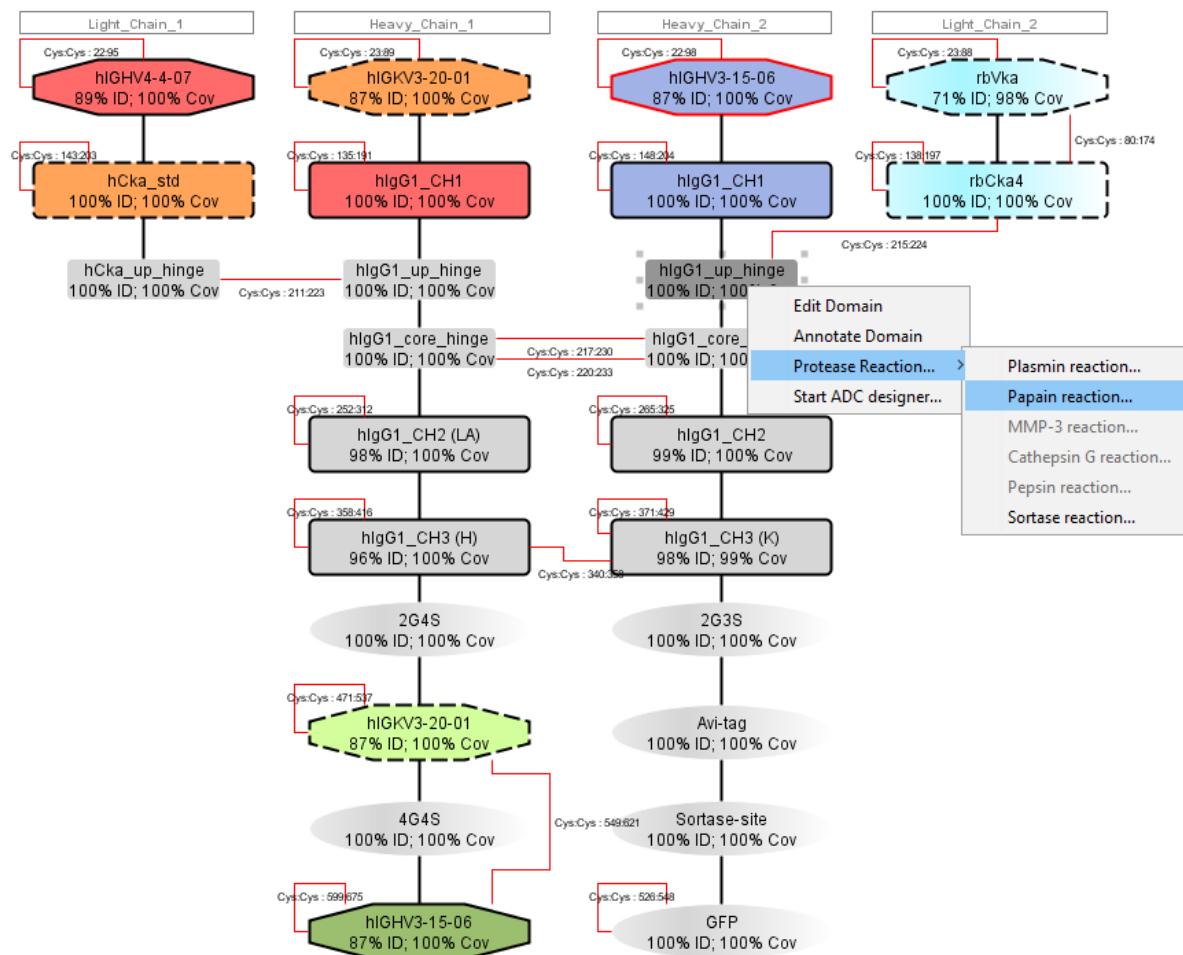
Clicking on the [Monomer Designer] in the [Registration] menu opens the "Monomer Designer".

Here, you can add new chemical monomers to the local monomer store. By clicking the [Save] button the monomer will be stored to the local monomer store.

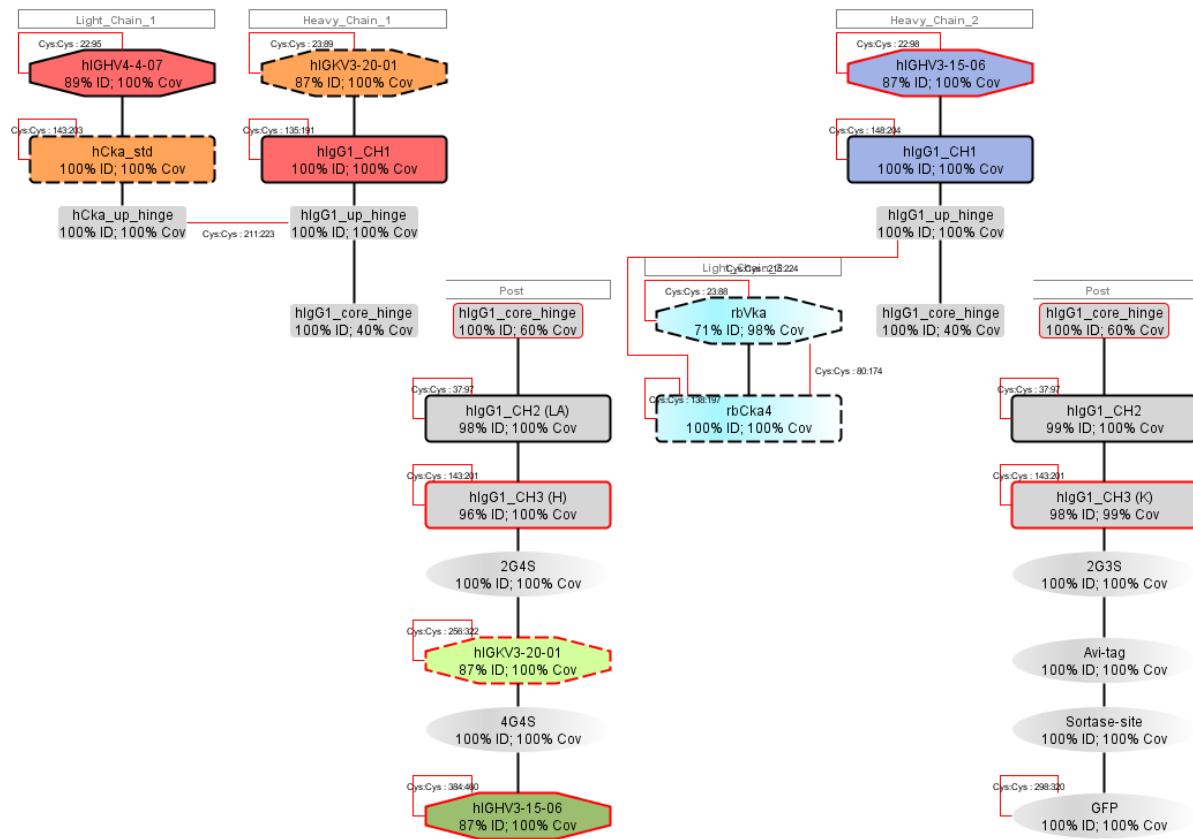


2.5.4 Domain Protease Reaction

In order to cleave peptides of your antibody, right-click any domain of a cleavable peptide and choose [Protease Reaction]. Select your desired protease.

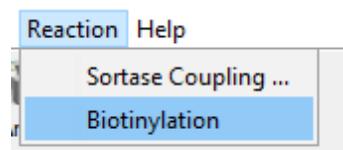


All chains matching the protease cleavage motif will be cleaved accordingly.

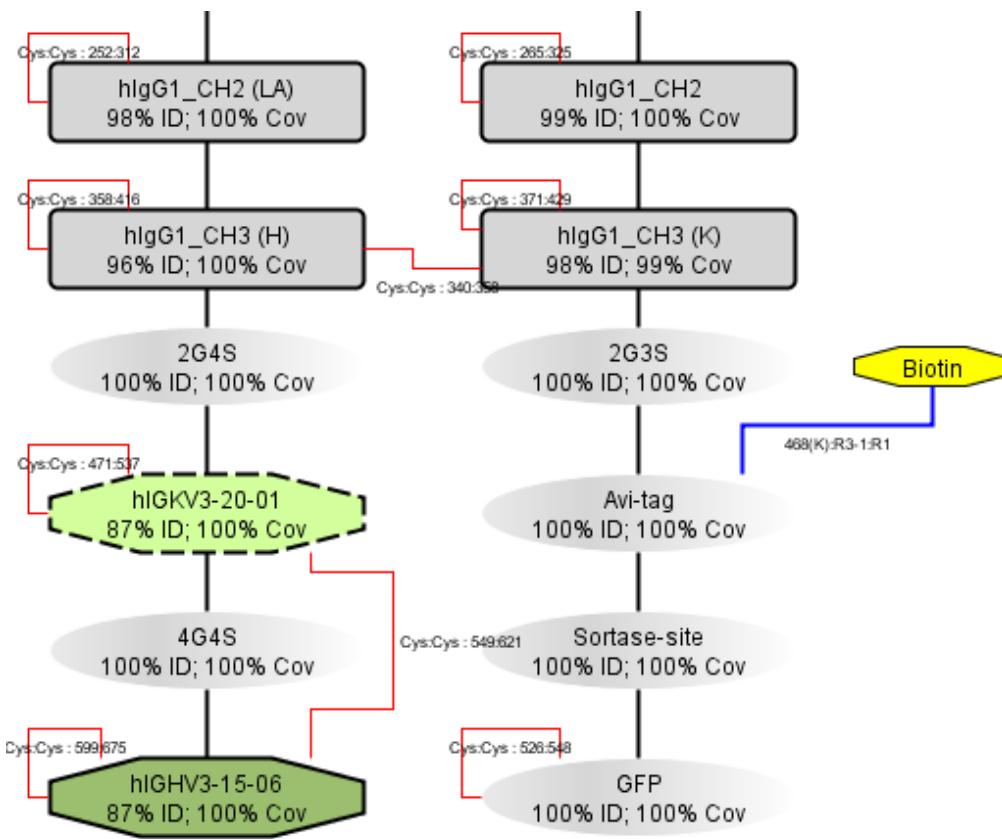


2.5.5 Biotinylation

Biotinylation is done by clicking on the [Biotinylation] button in the [Reaction] menu.



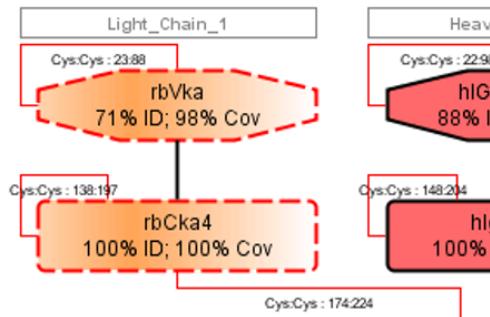
Biotin is attached to all domains called Avi-tag.



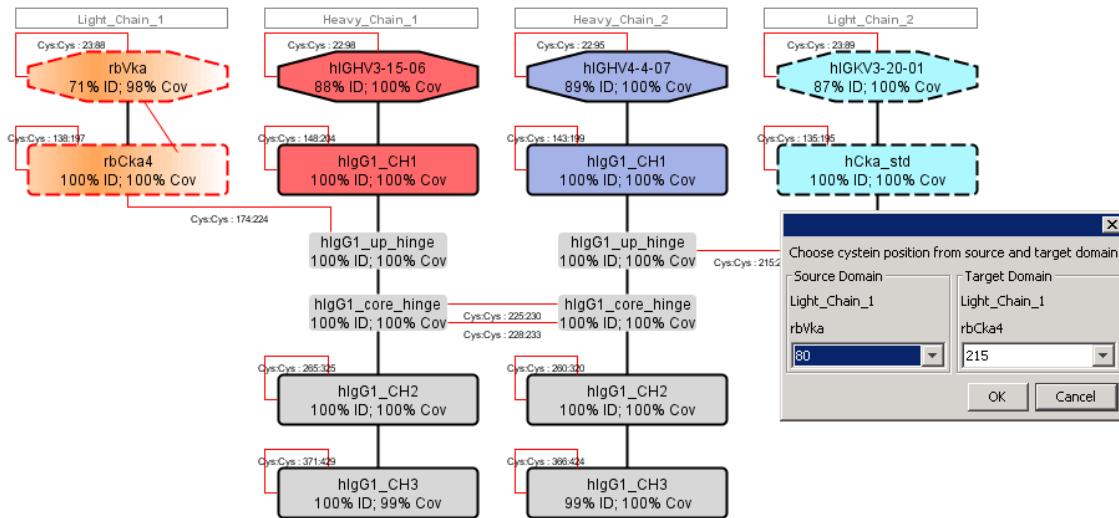
2.5.6 Connect domains manually

Creating a Cys-Cys bond is done by clicking on a domain, keeping the (left) mouse-button pressed and dragging to the domain you want to connect to.

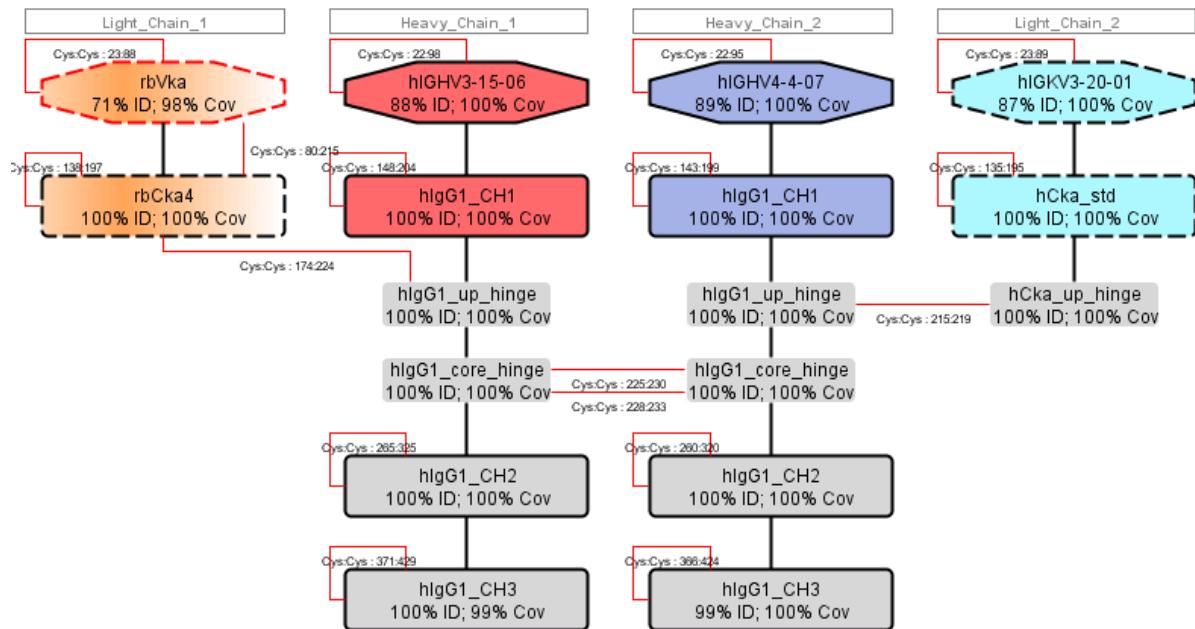
Thus click on the upper left domain of the test antibody and connect it with the domain below to get an additional (rabbit specific) Cys-Cys bond of these two domains. Please note: this is another example antibody than the one we showed above, see "sample\mixing test antibody.fa" from your HELMAntibodyEditor folder.



This opens a window inviting you to choose the cysteine position from source and target domain.



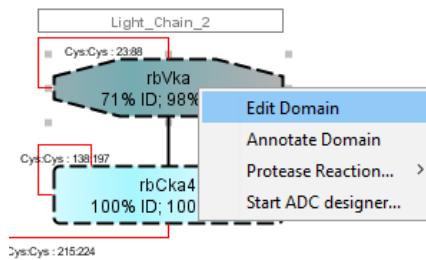
If both domains have only one cystein position, this window will not appear.
Click [OK] to result in :



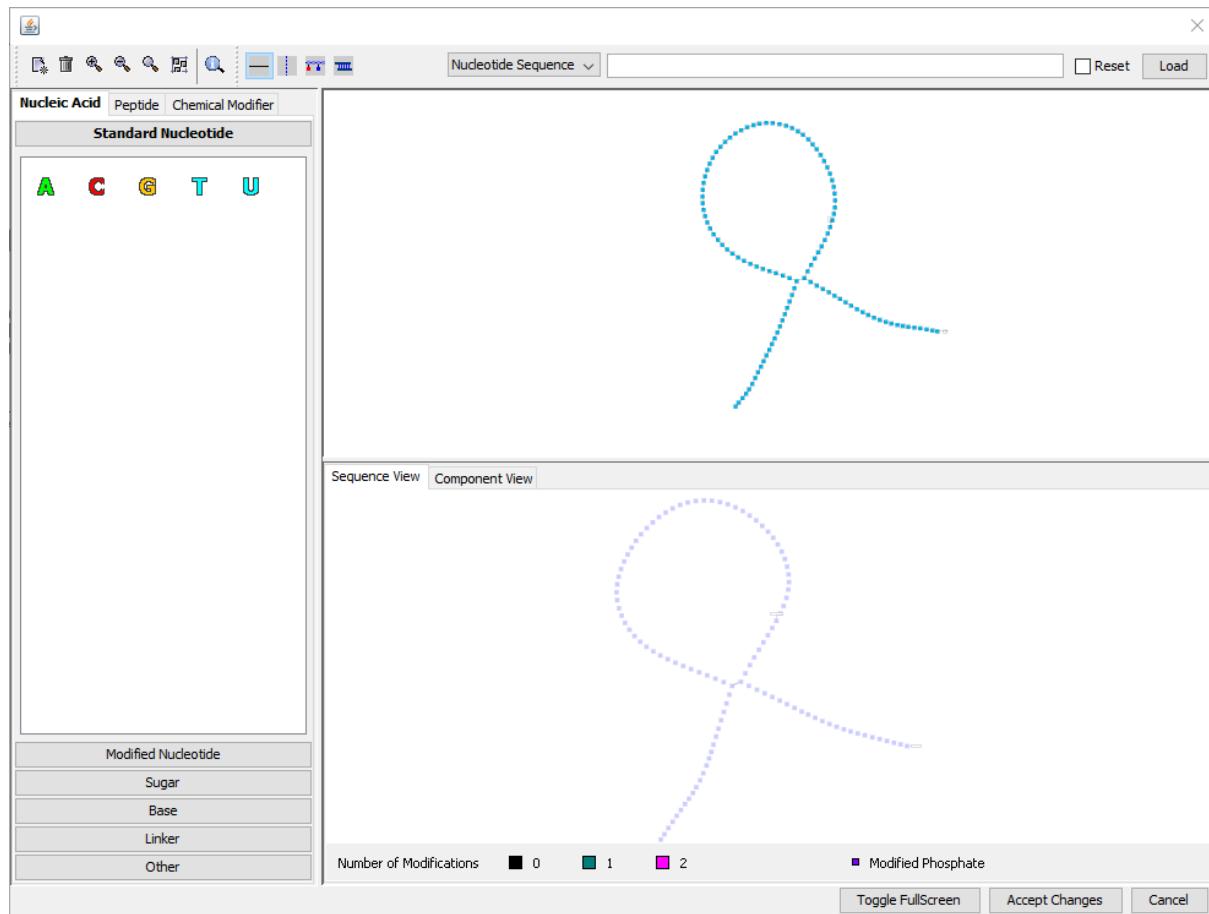
Connecting the constant light domains with the hinge regions results in a registerable antibody. The paired Fv regions are now marked by a common background color.

2.5.7 Edit domains in the HELM Editor

In order to chemically modify domains of your antibody, e.g. to add payloads, right-click the domain of your interest and choose [Edit Domain].

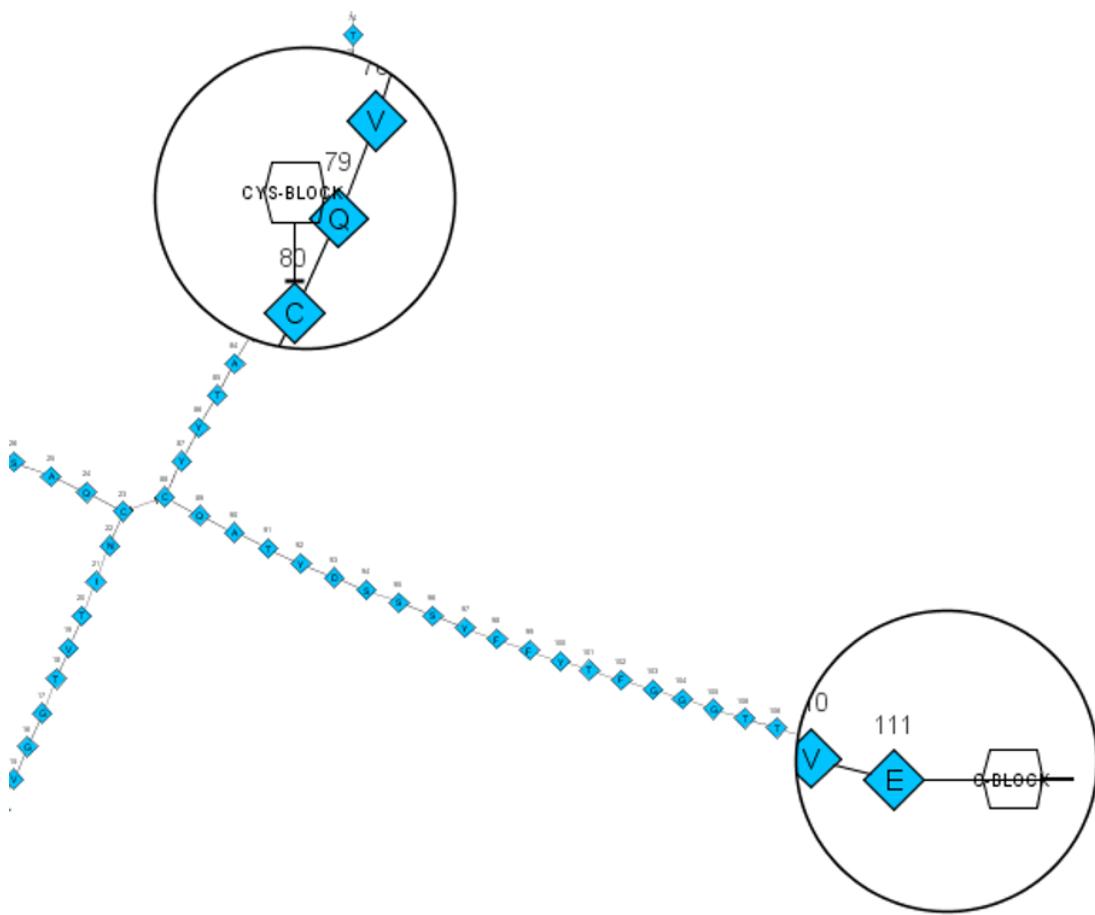


This opens the *HELM Editor*:



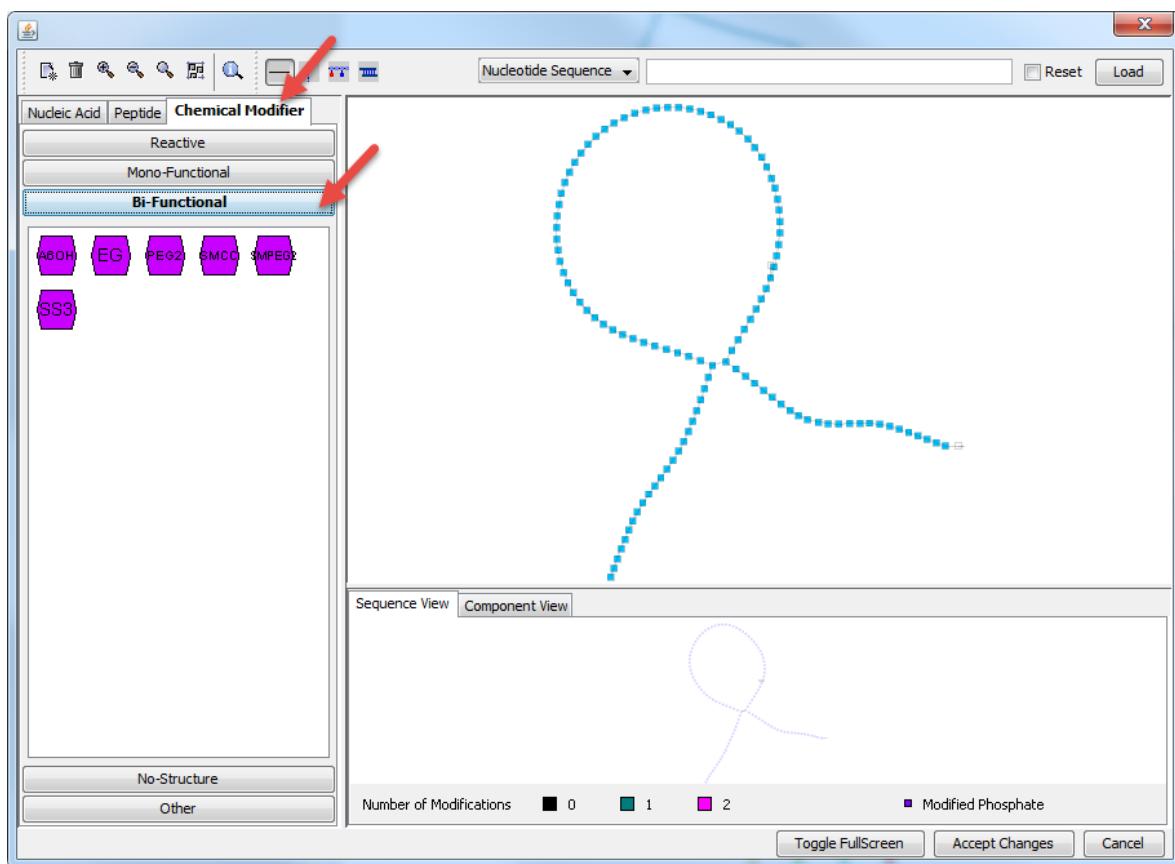
Concerning the sample antibody, the two Cys residues that connect to other domains are blocked:

- The domain C-terminus connected to the N-terminus of the "rbCka4" domain
- and the additional bridge into the "rbCka4" domain.

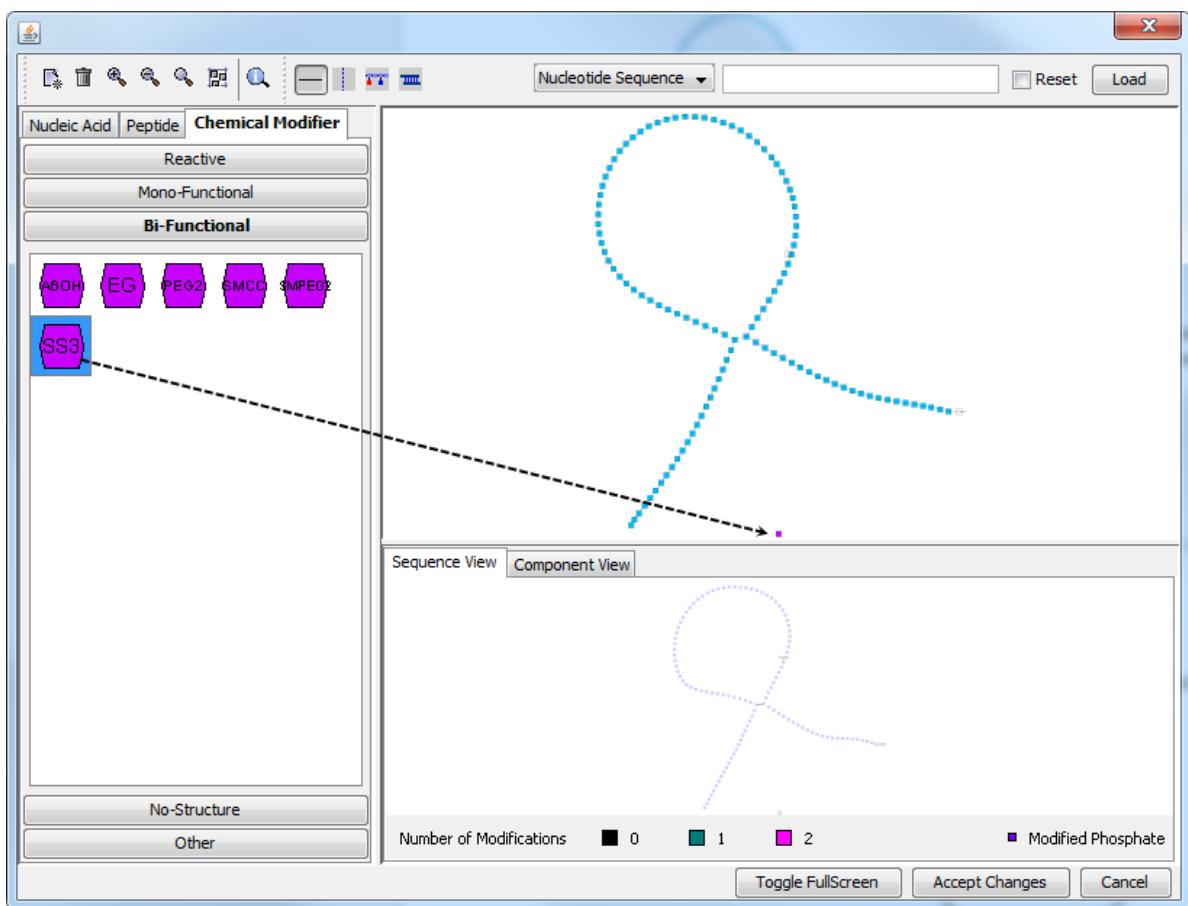


As an example we add a chemical linker and connect the nucleotide sequence GATC to the antibody:

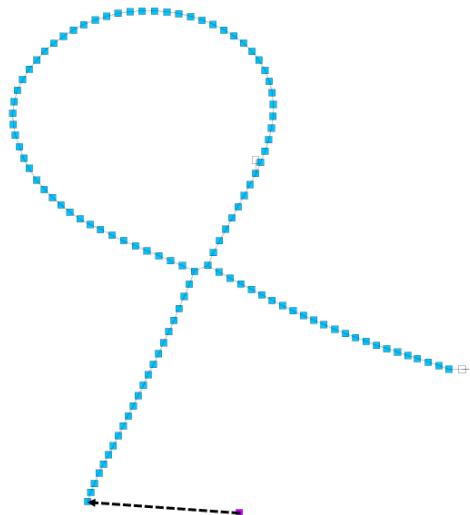
Choose [Chemical Modifier] from the monomer toolbar in the left of the *HELM Editor* and click the tab [Bi-Functional].



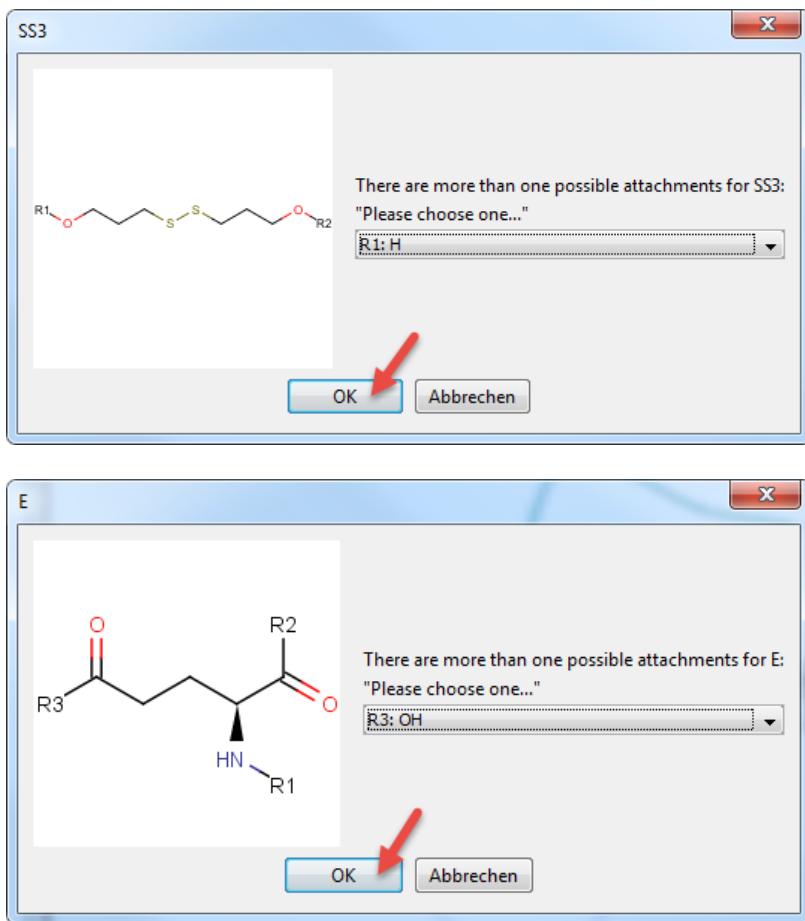
Select "SS3" and drag it onto the canvas.



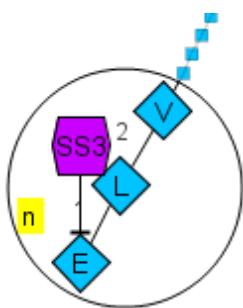
Connect the monomer to the free N-terminus by clicking on it, keeping the (left) mouse-button pressed and dragging to the N-terminus.



Choose the appropriate connection points (or just click [OK] if the correct R-group was preselected) in the dialogue windows for specification of the R-groups involved in this connection.

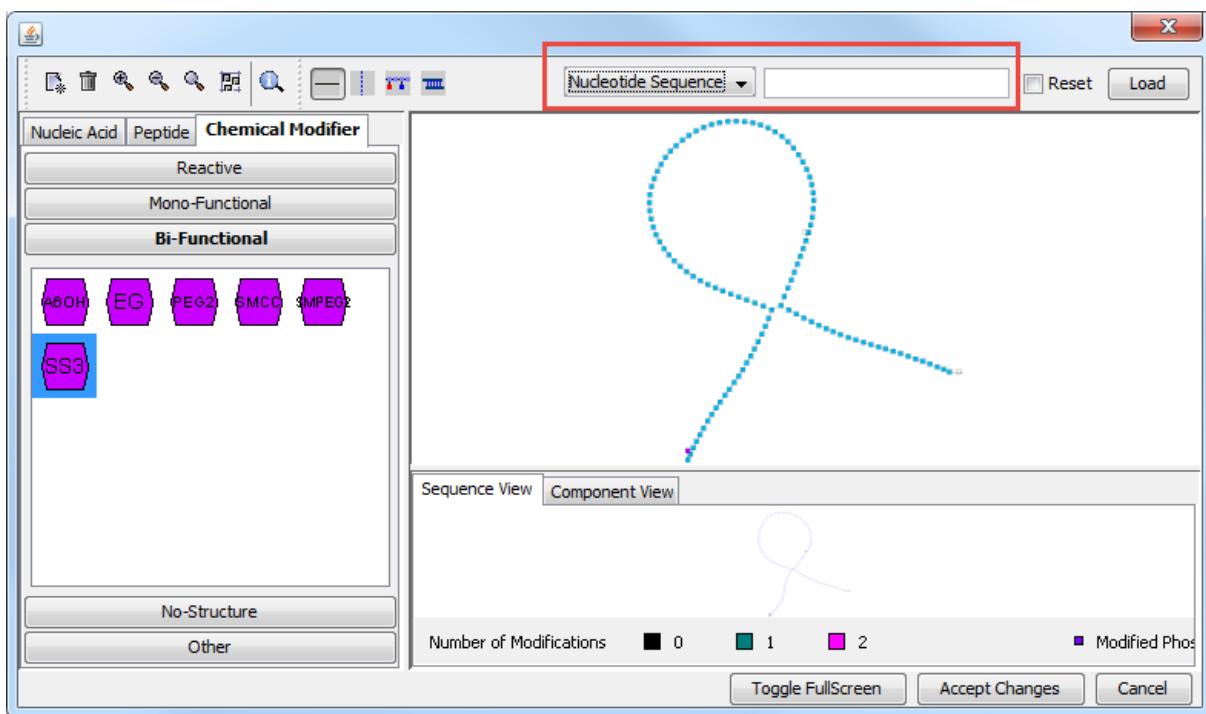


This will lead to:

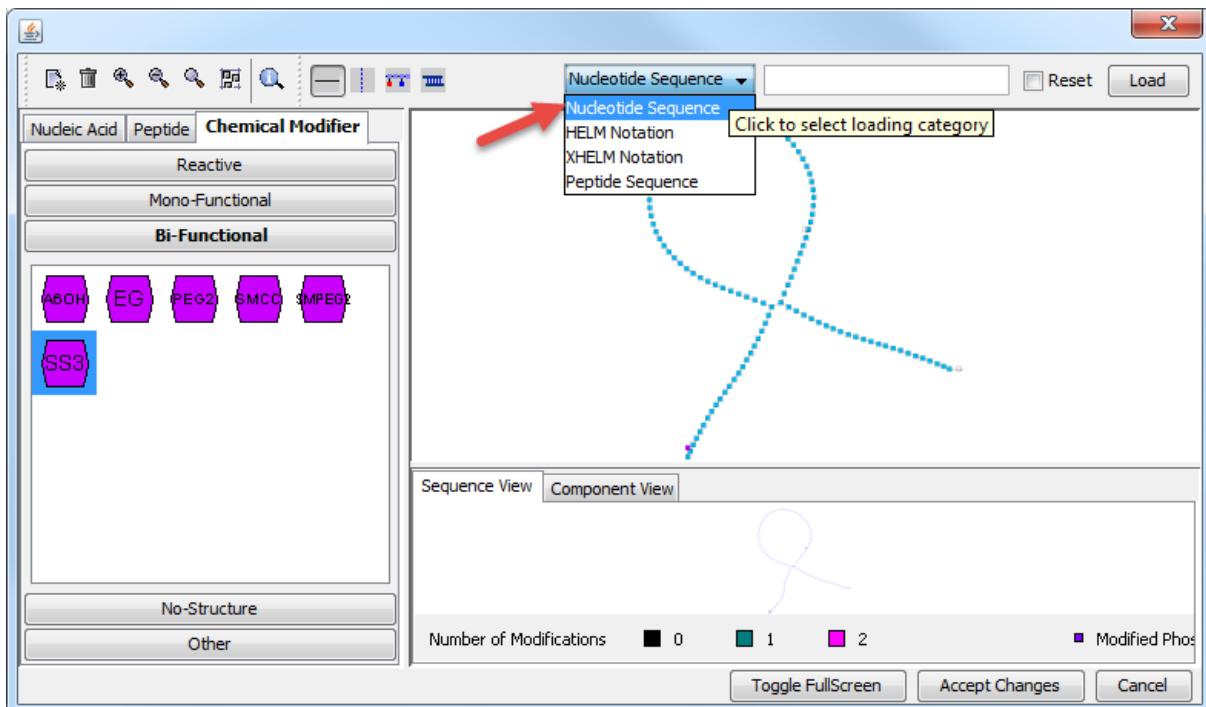


Now the nucleotide sequence "GATC" is loaded onto the canvas.

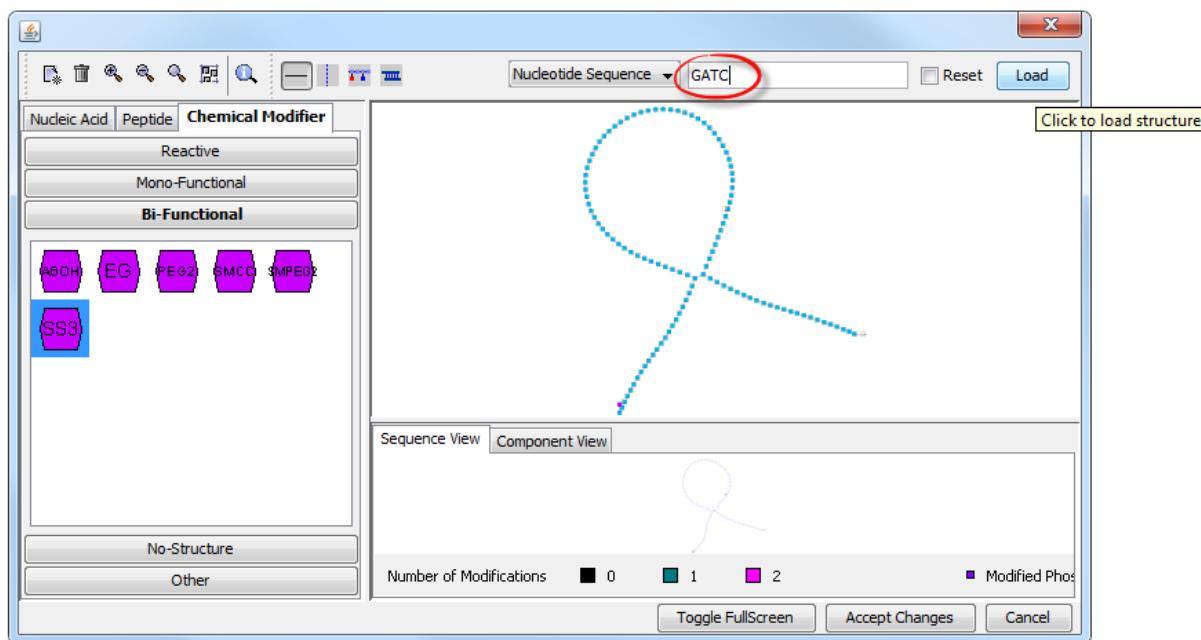
Use the input field at the top of the window:



Ensure that you selected the desired loading category.

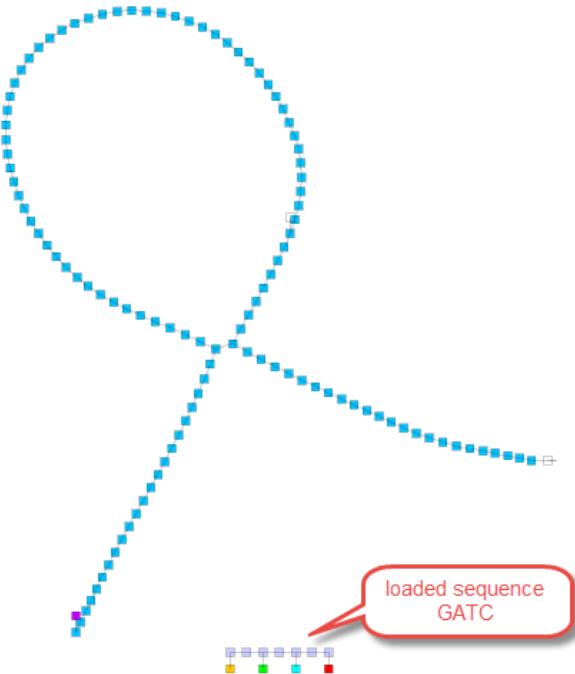


Type the nucleotide sequence "GATC" and press [Load] to load it onto the canvas.

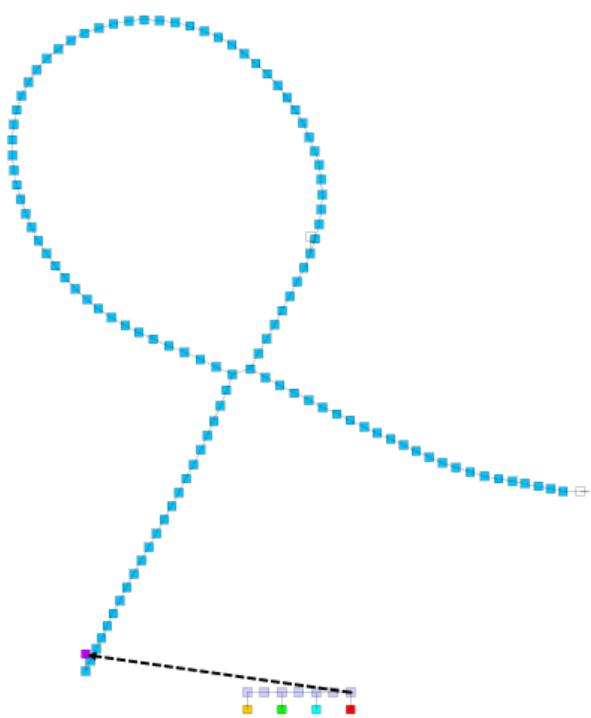


Alternatively you can use the monomer toolbar. Select [Nucleic Acid] and click the tab [Standard Nucleotide]. Drag the nucleotides onto the canvas and connect them manually.

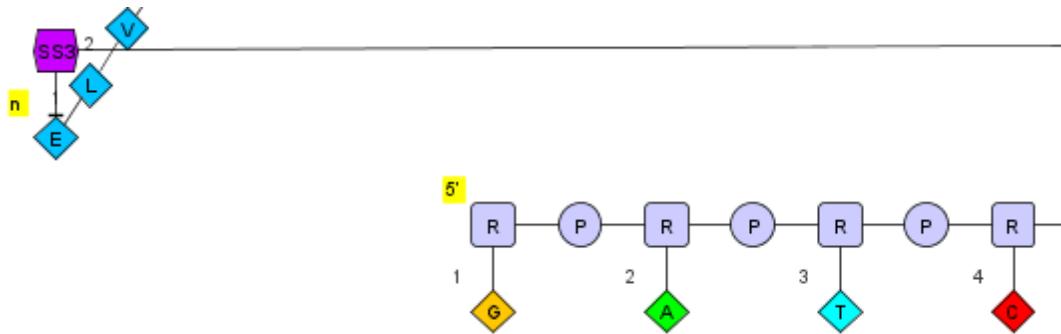
You will result in:



Connect the loaded nucleotide sequence to the "SS3".

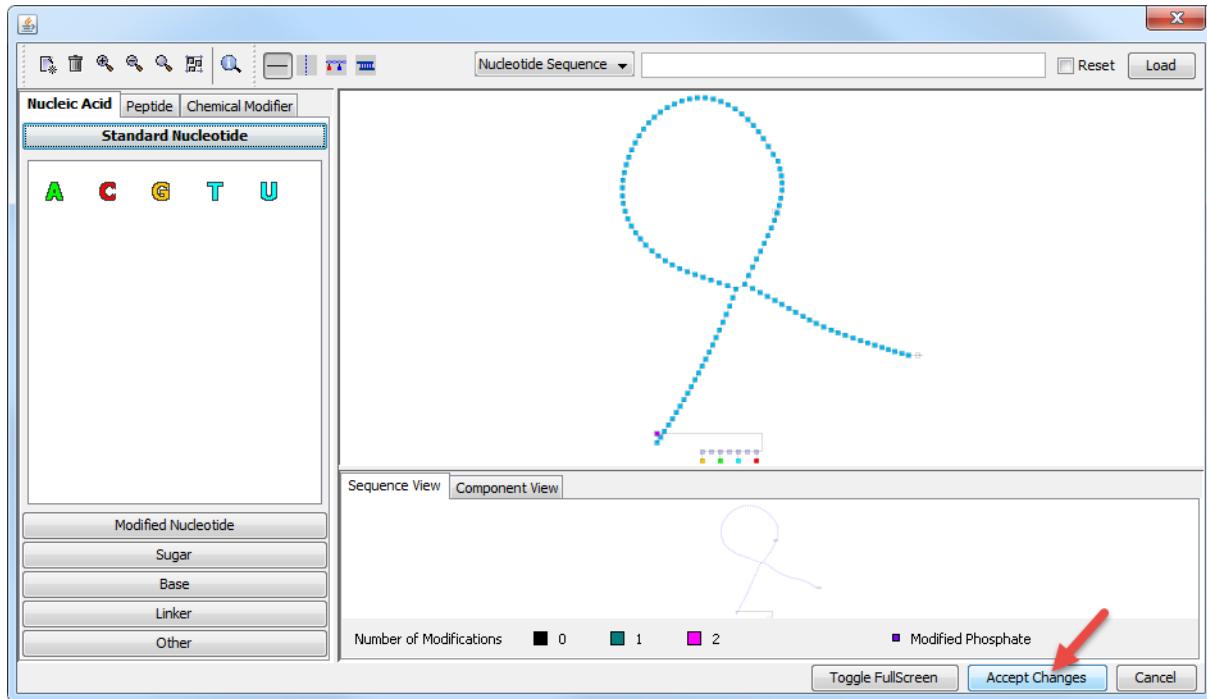


The nucleotide sequence is now added:

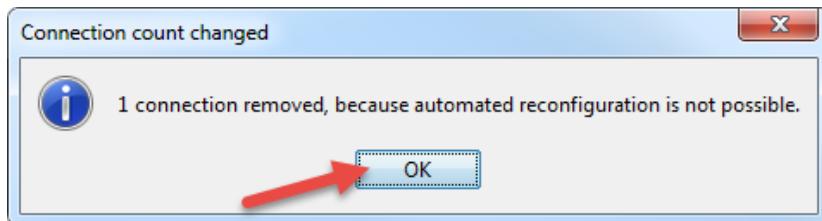


Note that you can also add peptides as well as CHEM payloads.

Click [Accept Changes] to return the modified domain to the *HELM Antibody Editor*.

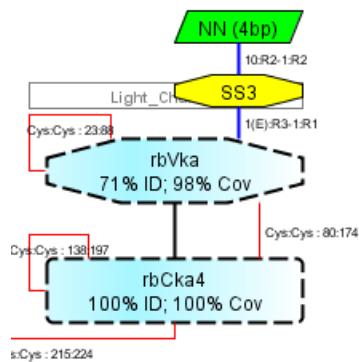


A window opens that indicates that the Cys-Cys bridge between the domains "rbVka" and "rbCka4" had to be removed for technical reasons and thus would have to be re-drawn.

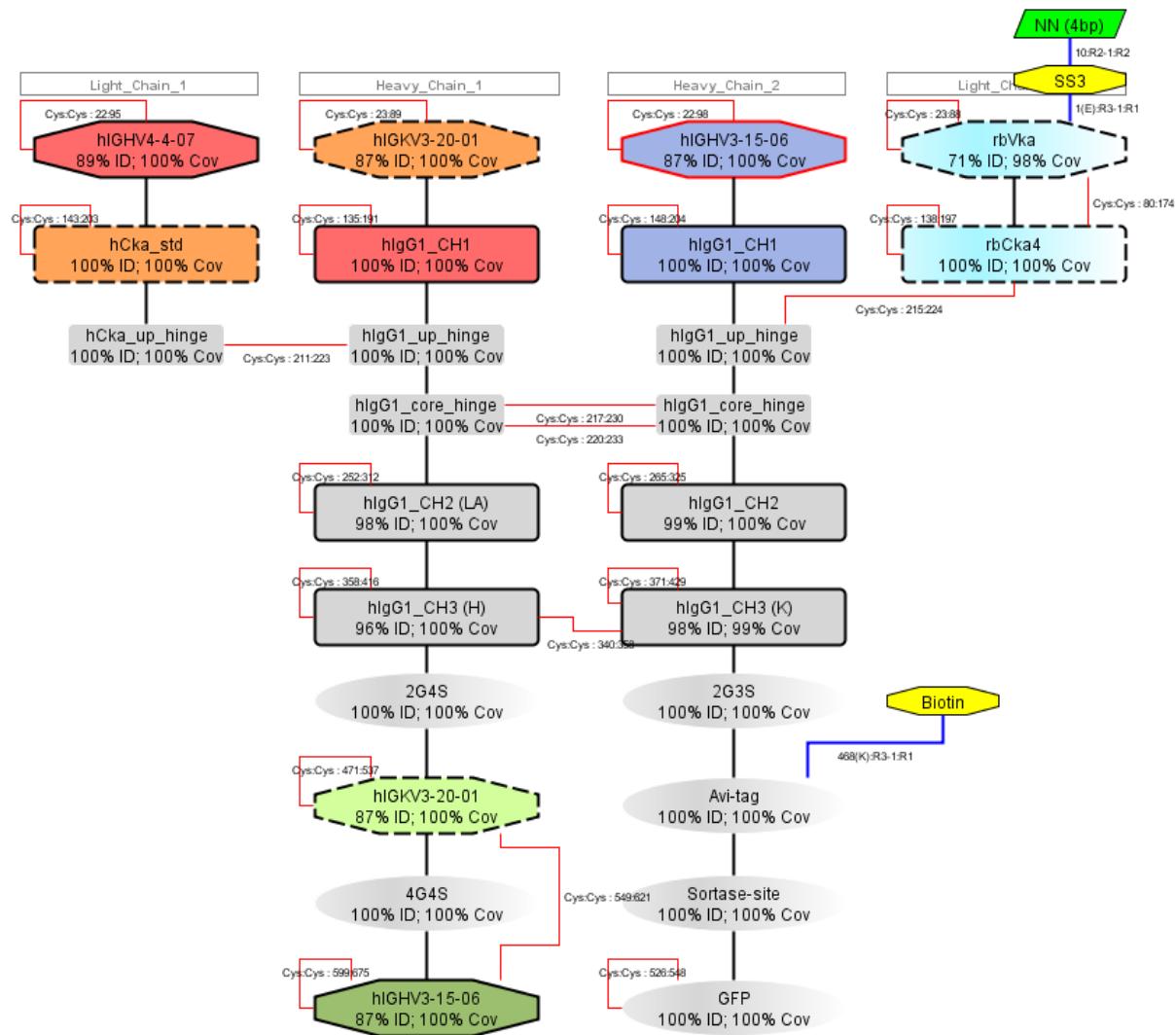


Click [OK] to get your modified antibody.

Re-draw the removed Cys-Cys bridge between "rbVka" and "rbCka4":

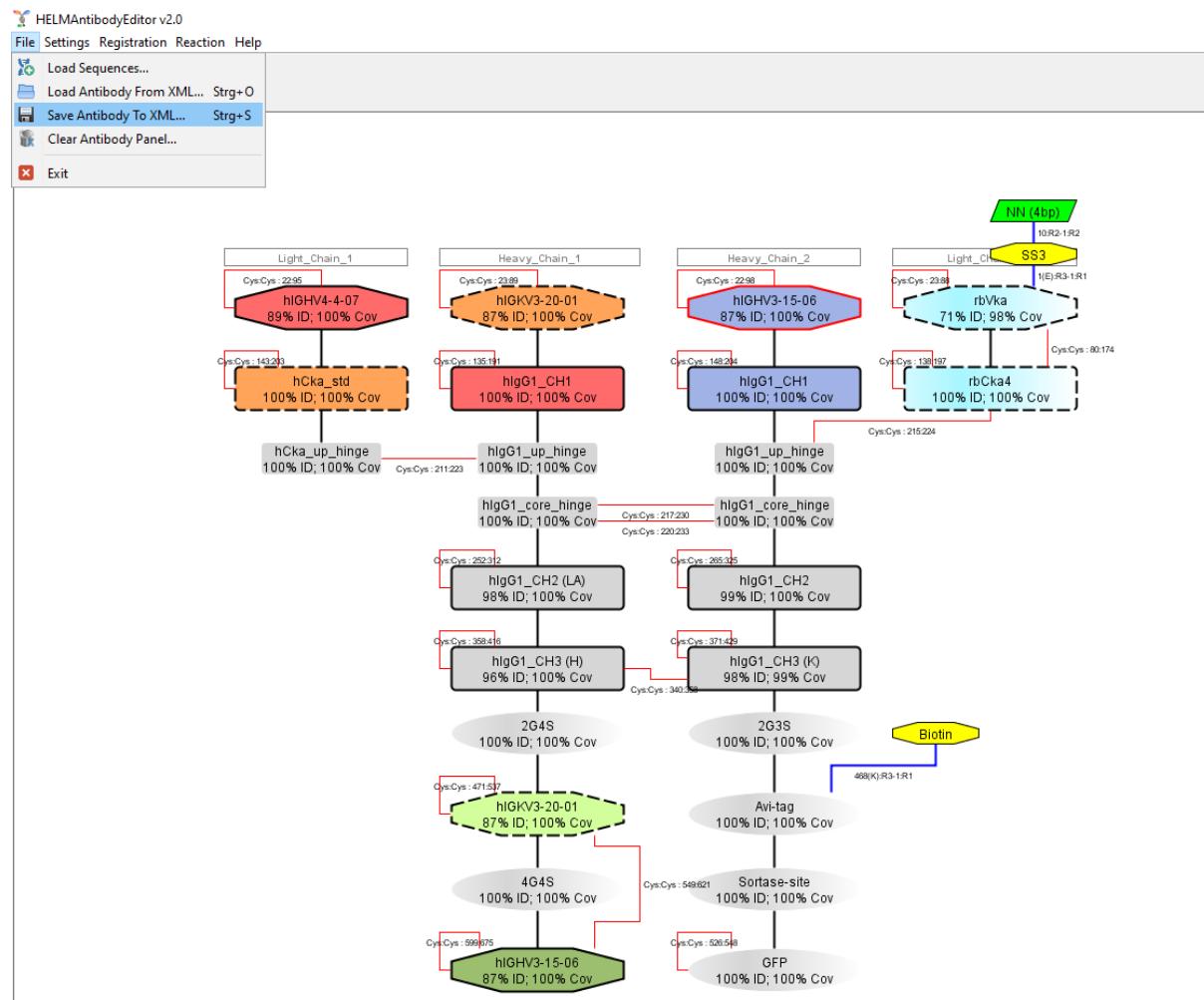


Once done this is your final molecule ready for registration:



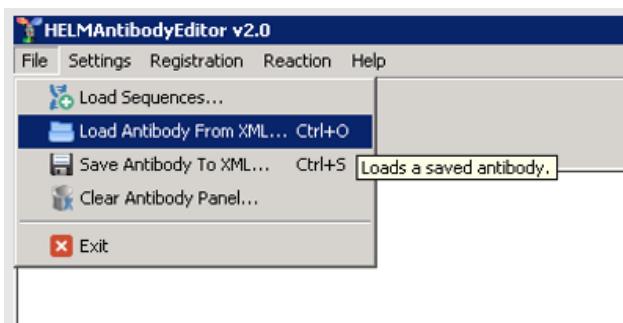
2.6 Save final antibody

To save your antibody to XML file, click [Save Antibody to XML...] under the menu [File].



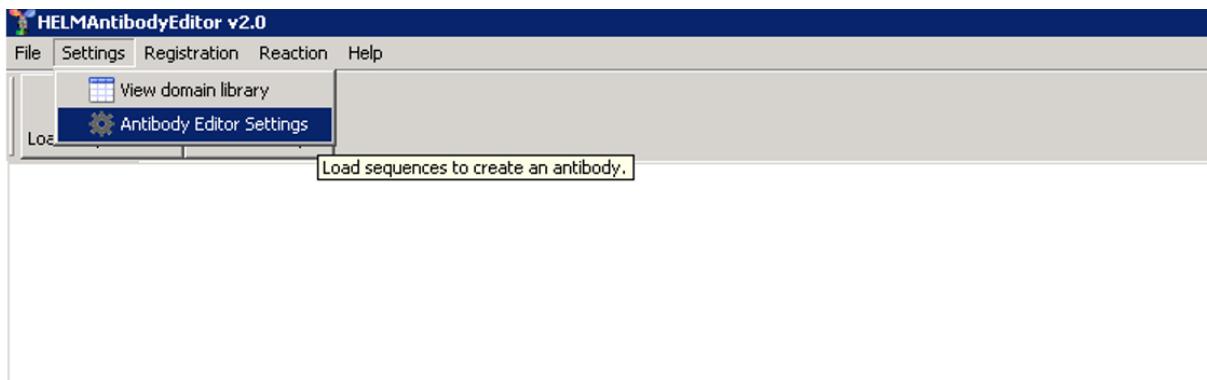
2.7 Load registered antibody

To load your antibody from XML file, click [Load Antibody from XML...] under the menu [File].

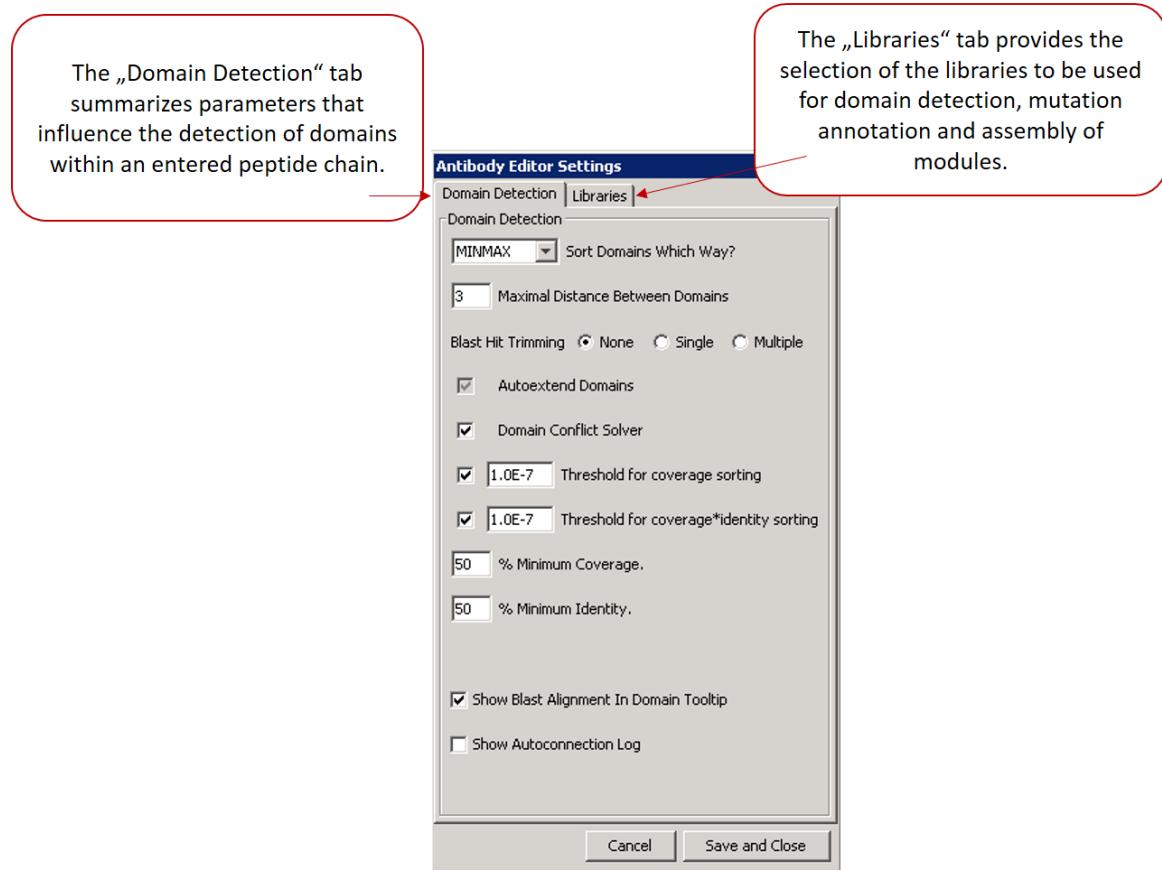


3 Antibody Editor Settings

To open the HELM Antibody Editor settings click [Settings] in the main menu and choose [Antibody Editor Setting].



The "Antibody Editor Settings" contain two tabs:

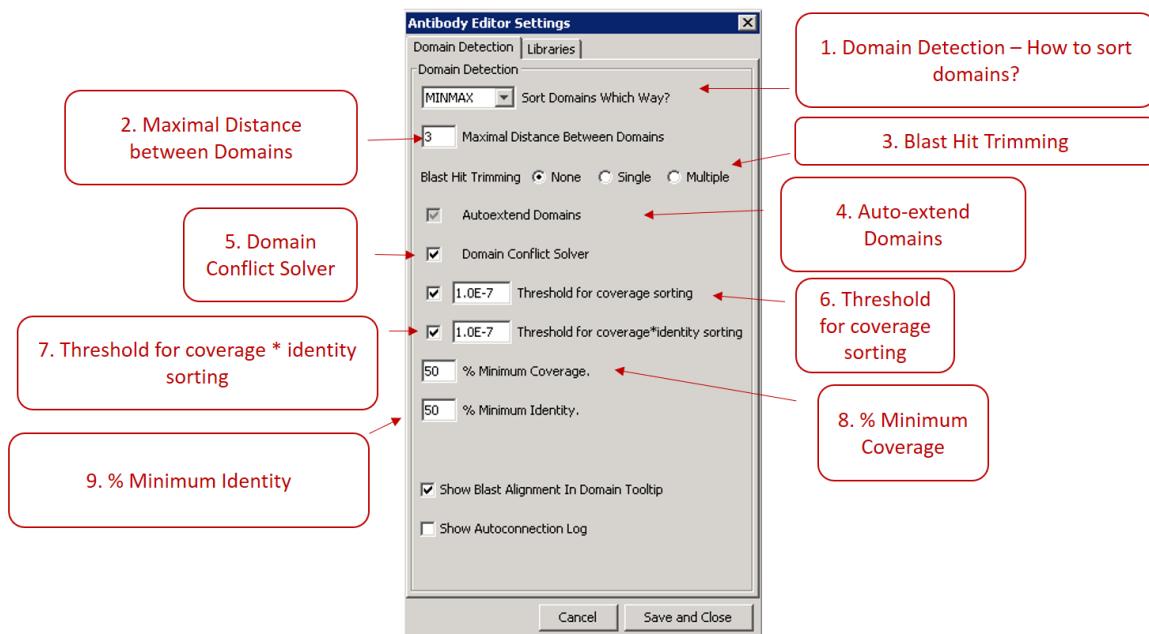


3.1 Domain Detection Settings

To understand the Domain Detection Settings first the domain detection procedure has to be briefly described:

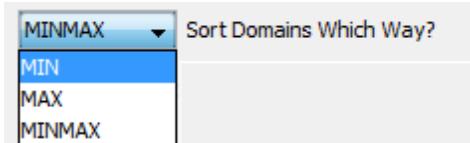
The entered peptide chains are chopped into domains using BLAST versus the domain library. Domain hits are filtered for their %identity and %coverage, sorted and finally assembled. In the Domain Detection Window the entered chains are listed below each other and the domain hits represented next to each other in rows (s.above). These domain hits are sorted depending on different criteria that depend on the BLAST e-value (given in brackets after the domain name) according to options in the Domain Detection settings.

The default (and recommended) window contains nine options:



1. Domain Detection - How to sort domains?

You can choose between three alternative algorithms:



MIN finds domains as small/ granular as possible.

MAX finds domains as large as possible.

MINMAX combines the strengths of MIN and MAX: It maps the filtered and sorted BLAST hits to the entered peptide sequence. Any domain with a lower sorting rank is either also mapped onto the entered peptide sequence as space allows (max. one residue overlap accepted to already mapped domains) or otherwise is placed below an already mapped domain and thus assigned to its given position. Thus the best (top sorted) hits determines the position and size of the detected domains.

2. Maximal Distance between Domains

If BLAST can't annotate stretches between domains these "domains" are labeled "NN (...AAs)". In case the number of such residues is equal or smaller than this parameter this inter-domain stretch is not marked as a domain as long as it doesn't contain a Cys.

3. Blast Hit Trimming

This allows to automatically remove mismatch (case: single) or mismatches (case: multiple) at the end of the blast alignments. There are three options: None (Default), Single or Multiple. It is recommended not to change this setting.

4. Auto-extend Domains

In case a given (entered) domain has terminal differences to the best matching domain from the domain library, BLAST will fail to return the full alignment. If this option is checked the given domain is extended beyond the BLAST hit until either the length of the library domain is reached or other detected domains would cause an overlap conflict. This is useful to extend especially variable domains to their full size as the given antibodies might well diverge from the sequences in the domain library.

5. Domain Conflict Solver

In case two domains share one common residue at their ends BLAST might return overlapping domains that are marked by a red border of the conflicting domains in the Domain Detection Window. If the Domain Conflict Solver option is checked the software assigns this one common residue to that domain with the higher priority (i.e. lower domain_priority number, s. chapter 4.2). This option is quite useful to support batch registration of biologicals as it avoids further user interaction in case of conflicting domains.

6. Threshold for coverage sorting

BLAST hits with an e-value lower than this parameter are significant and sorted for their e-value. BLAST hits with an e-value higher than this parameter are often linkers (e.g. 2G3S) and just sorted for their coverage (1st) and e-value (2nd) to select the best hit.

7. Threshold for coverage*identity sorting

BLAST hits with an e-value lower than this parameter are highly significant (e.g. full domains) and sorted for their coverage*identity. This helps to select the better match from close consensus sequences independent of their length – an attribute that influences the BLAST e-value in an non-controllable manner.

8. % Minimum Coverage

Filters for BLAST hits equal or above % coverage specified by this parameter. "%Coverage" relates to the extent to which a domain from the domain library is used to annotate a sequence stretch of the input sequence. This parameter shall remove small stretches of domains from the library to be matched to the input sequence.

9. % Minimum identity

Filters for BLAST hits equal or above % identity specified by this parameter.

10. Show Blast Alignment in Domain Tooltip

The BLAST alignment will be shown in the "Antibody Domain Found" window for every domain when the mouse is moved over the domains.

11. Show Autoconnection Log

The autoconnector rules applied are shown when the antibody has been automatically assembled. This is useful for administrators but less informative to users.

As detailed two parameters influence the sorting of hits. The following list illustrates the scale from low (significant) to high (not significant) e-values:

Coverage*Identity sorting (for large domains), e-value if otherwise equal

...

Threshold for coverage*identity sorting : to be applied only on large domains to focus on them

Alternatively this value can be set equal or close to "Threshold for coverage sorting" with the effect that large and small domains are treated the same way and thus the entered peptide chain might be chopped into a set of smaller domains instead of one modified larger one. This has to be evaluated for each user group/ company independently.

...

e-Value sorting (for small domains)

...

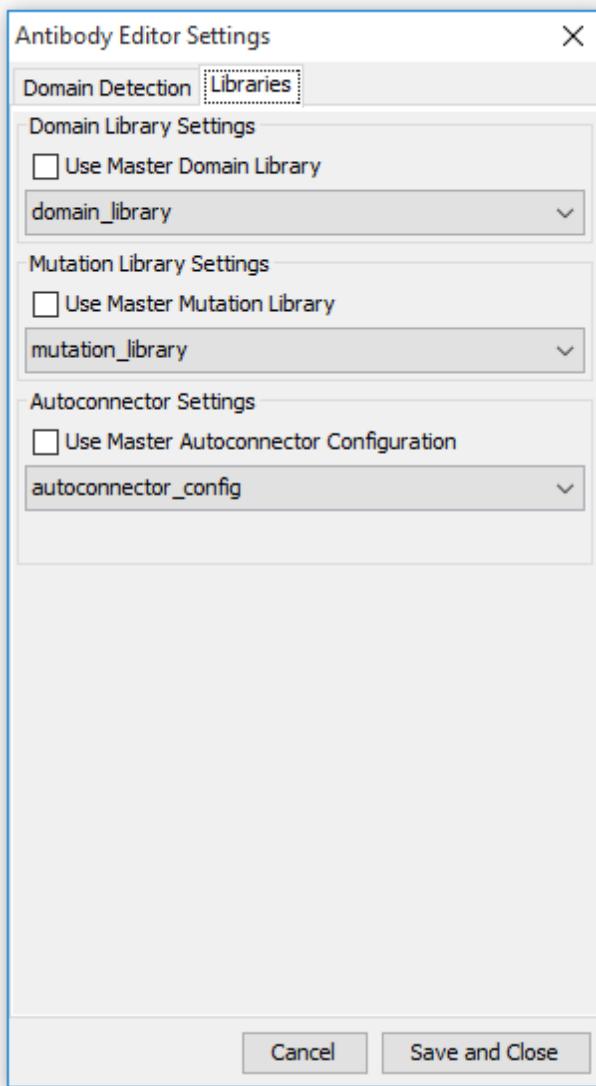
Threshold for coverage sorting

...

Coverage / e-value sorting (for e.g. xGyS linkers), e-value if otherwise equal

3.2 Libraries

The tab [Libraries] allows to select the libraries. In a company installation the three check boxes [Use Master Domain Library], [Use Master Mutation Library] and [Use Master Autoconnector Configuration] select for a remote database system. As a starter the public version on GitHub contains a SQLite library file in config\config.db3 containing different instances of these libraries. See details in Chapter 5. In the GitHub version there is only one instance for each library that is set as default: See also How to instructions in resources directory to implement your own master libraries.



4 Antibody Editing Window

Note: The legend (s. [chapter 1.2.5](#)), the Details Box (s. [chapter 2.4](#)), manual connection of domains (s. [chapter 2.5.6](#)) and adding oligos, peptides and/or chemicals to domains (s. [chapter 2.5.7](#)) are explained in the previous sections.

From the Domain Detection Window the domains are analyzed for mutations and annotated accordingly.

4.1 Cys-Cys bond creation

Next the antibody is assembled with as many Cys-Cys bonds automatically created as possible according to the following steps:

- Create intra-domain Cys-Cys bonds as defined by the [Domain library](#)
- Create hinge to hinge Cys-Cys bonds (hard coded)
 - Only if same subtype, same number of Cys, sequences might differ
- Create hinge to constant light chains Cys-Cys bonds (hard coded)
 - Only if HC or LC or both are fully identical (assuming 2xHC + 2xLC)
- Create mutation defined inter-chain Cys-Cys bonds as defined by the [Mutation library](#)
 - E.g. knob into hole
- Create intra-chain / inter-chain Cys-Cys bonds as defined by the [Autoconnector rule set](#)
 - E.g. scFv, scFab

4.2 Canvas options

All bonds (thin red lines) can be selected, the position of their anchor points be adjusted and even deleted.

Domains can be right-clicked providing the following options:

- **Edit Domain:** s. [chapter 2.5.7 Edit domains in the HELM Editor](#).
- **Annotate Domain:** re-annotate domain (by BLAST of just this domain)

A right-click on the canvas opens a menu with the following options:

- **Back to Domain Recognition:** opens the Domain Detection Window. This can be used to restore the automatically generated antibody assembly if needed.

- **Reset Layout:** resets layout to yFiles defaults
- **Reset Cystein Bridges (+Layout):** resets layout and removes all intra-domain Cys-Cys bonds as defined by the domain library
- **Show HELM (version 1) :** shows the always up-to-date HELM code in the specified version for the complete molecule on the canvas

5 Libraries

The HELM Antibody Editor uses four libraries: the domain library, mutation library, autoconnector rule set and protease library. They are provided as SQLite file in config\config.db3. With the HELM Antibody Editor alone, it is neither possible to enter additional domains, linkers, novel Fc mutations,etc. nor is it possible to extend the rules that you use to detect more complex antibody formats.

To perform modifications like this, we can recommend using <http://sourceforge.net/projects/sqliteman/> as an easy and standalone viewer / editor for SQLite database files (no installation required for Windows).

Please also refer to the document "Custom Master Libraries in HAbE v2".

5.1 Domain library

The domain library is used to identify and annotate the domains as building blocks of peptide chains. All additional domains, linkers, peptides, conjugate proteins that are used should be entered in this library. You can also see the domain library by clicking [Settings] [View domain library].

The domain library (only 1st 4 lines) looks like:

domain_pk	domain_name	domain_short_name	species	humanness	chain_type	domain_type
1	rblGLV6_signal_peptide	rblGLV6_SP	Rabbit (Oryctolagus cuniculus)	humanizable	lambda	signal_peptide
2	rblGLV5_signal_peptide	rblGLV5_SP1	Rabbit (Oryctolagus cuniculus)	humanizable	lambda	signal_peptide
3	rblGLV5_signal_peptide	rblGLV5_SP	Rabbit (Oryctolagus cuniculus)	humanizable	lambda	signal_peptide
4	rblGLV4_signal_peptide	rblGLV4_SP	Rabbit (Oryctolagus cuniculus)	humanizable	lambda	signal_peptide

domain_sequence	domain_cys_pattern	domain_comment	domain_sawi	domain_usage	is_active	domain_priority	allowed_free_cys
MAWAPLLLTVAHCTGSWA				GENERAL	1	8	1
VTLAAPLLSTLTSEISPVPS				GENERAL	1	8	0
MAWAPLLLLLSHCTGSLS				GENERAL	1	8	1
MACTPLLLLTLQCTGSLS				GENERAL	1	8	2

The following columns are included:

- **domain_pk**: used as internal primary key.
- **domain_name**: full description of the domain. Name that is referred to in the mutation library.
- **domain_short_name**: name used in BLAST and in the GUI.
- **species**: human, mouse, rat, rabbit or empty / “-” for e.g. linkers
- **humanness**: human, humanizable if not human yet, non-human if not humanizable as e.g. linkers
- **chain_type**: heavy, kappa, lambda, empty / “-”
- **domain_type**: variable, hinge, constant, empty / “-”
- **domain_sequence**: amino acid sequence
- **domain_cys_pattern**: the pattern of Cys residues as ordinal numbers that pair within this domain, heavy chain domains might pair with light chains “LC” and light domains might pair with hinge “H”. Multiple bonds are separated by a comma.
- **domain_comment**: an arbitrary comment for your convenience.
- **domain_sawi**: not used
- **domain_usage**: You can use CORE or GENERAL for partitioning your domain library. Both will be

blasted separately and the results will be merged together to find the best hits according to the user-defined hit sorting.

- **is_active:** only entries with is_active=1 will be used by BLAST. 0 means ‘inactive’.
- **domain_priority:** In case of conflicting domains (s. chapter 4.2) the residue shared by 2 domains is assigned to the domain with the higher priority (i.e. lower domain_priority value).
- **allowed_free_sys:** number of allowed free cysteins. It might influences the color of domain border.

The domain library contains the variable and constant antibody domains for the species human, mouse, rat, and rabbit. The **variable regions** (heavy, kappa, lambda) are provided as the individual V-element germlines taken from IMGT combined with a CDR3 space holder (XXX...XXX) and a common J-element.

The **constant domains** are based on Swissprot (that by itself has received the Ig constant regions in 2008 from IMGT) and IMGT and are chopped into domains like CH1, hinge, CH2, and CH3.

- Lefranc, M-P. Immunoglobulins: 25 years of Immunoinformatics and IMGT-ONTOLOGY. *Biomolecules*. 2014, 4, 1102-1139; doi:10.3390/biom4041102 Open access, <http://www.mdpi.com/2218-273X/4/4/1102>
- Lefranc M-P, Giudicelli V, Duroux P, Jabado-Michaloud J, Folch G, Aouinti S, Carillon E, Duvergey H, Houles A, Paysan-Lafosse T, Hadi-Saljoqi S, Sasorith S, Lefranc G, Kossida S. IMGT®, the international ImMunoGeneTics information system® 25 years on. *Nucleic Acids Res.* 2014 Nov 5. pii: gku1056. [Epub ahead of print] PMID: 25378316 Free Article: <http://nar.oxfordjournals.org/content/early/2014/11/05/nar.gku1056.abstract>
- The UniProt Consortium
Activities at the Universal Protein Resource (UniProt)
[Nucleic Acids Res. 42: D191-D198 \(2014\).](http://nucleic.acids.res.42:D191-D198.(2014).)

5.2 Mutation library

The mutation library contains all known deviations from the domains available in the domain library. Enter variations as e.g. mutations and allotypes of such domains to be identified. Mutations are grouped into sets (identical “mutation_name”) that will only be recognized as such if all the listed mutations of a Mutation group are found - otherwise the mutations are listed as multiple unknown mutations.

The mutation library (only 1st 4 lines) looks like:

mutation_pk	domain_name	mutation_name	mutation_short_name	mutation_num_var	mutation_position_in_domain
1	IGHG4_HINGE_HUMAN	SPLE	P+E	2	10
2	IGHG4_CH2_HUMAN	SPLE	P+E	2	5
3	IGHG1_CH3_HUMAN	G1m3_nG1m1	A	3	18
4	IGHG1_CH3_HUMAN	G1m3_nG1m1	A	3	16

mutation_kabat_index_eu	mutation_original_aa	mutation_new_aa	mutation_visible	mutation_connection_group
228	S	P	1	
235	L	E	1	
358	L	M	0	
356	D	E	0	

The following columns are included:

- **mutation_pk**: used as internal primary key.
- **domain_name**: name of the domain as listed in the domain library under “domain_name”
- **mutation_name**: full name of the mutation (group)
- **mutation_short_name / mutation_visible**: name that is shown in brackets within the domain as shown on the canvas if “mutation_visible” is set to 1 else the mutation is not shown there but just in the details box to the right
- **mutation_num_var**: number of mutations ('variation') belonging to one mutation group
- **mutation_position_in_domain**: absolute position of the mutation within the domain
- **mutation_kabat_index_eu**: position of the mutation according to Kabat EU Index (only added for documentation; not used in the HELM Antibody Editor yet)
- **mutation_original_aa / mutation_new_aa**: original and mutated amino acid
- **mutation visible**: If it is on 1, the mutation short name will be shown in brackets after the name of the visualized domain
- **mutation_connection_group**: identical strings (not essentially numbers) indicate Cys residues to be connected if both Cys are found, and only if all mutations of the corresponding mutation group are found.

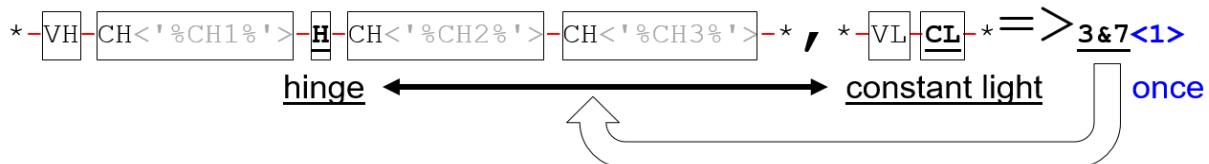
5.3 Autoconnector rule set

Rules recognize modules like e.g. scFv and scFab and define the Cys bond pattern between domains of such modules or even beyond those modules. They are provided as text in a separate table (with line numbers to define their sequence) that look like (only 1st 10 lines):

line_pk	line
10	#rabbit extra Cys-cys bridge between VH and Vkappa
20	rbVL-CL-* rbIGKV%->-CL<'rbCka%'->-*=>1&2#2<n>
30	rbCH1-H-* CH<'rbIgG_CH1'->-H<'rbIgG_hinge'->-*=>1#2&2#1<n>
40	#scFab connect V
50	scFab_VH-VL:-VH-CH<%CH1%>-X<%lgG1_up_hinge'->-X<?G?S'->-VL-CL<%Cka%_std%>-X<%Cka%_up_hinge'->-*=>1&5<n>
60	scFab_SP-VH-VL:-X<%IGV%_SP'->-VH-CH<%CH1%>-X<%lgG1_up_hinge'->-X<?G?S'->-VL-CL<%Cka%_std%>-X<%Cka%_up_hinge'->-*=>2&6<n>
70	scFab_-VH-VL:-VH-CH<%CH1%>-X<%lgG1_up_hinge'->-X<?G?S'->-VL-CL<%Cka%_std%>-X<%Cka%_up_hinge'->=>1&5<n>
80	scFab_VL-VH:-VL-CL<%Cka%_std%>-X<%Cka%_up_hinge'->-X<?G?S'->-VH-CH<%CH1%>-X<%lgG1_up_hinge'->-*=>1&5<n>
90	scFab_SP-VL-VH:-X<%IGV%_SP'->-VL-CL<%Cka%_std%>-X<%Cka%_up_hinge'->-X<?G?S'->-VH-CH<%CH1%>-X<%lgG1_up_hinge'->-*=>2&6<n>
100	scFab_-VL-VH:-VL-CL<%Cka%_std%>-X<%Cka%_up_hinge'->-X<?G?S'->-VH-CH<%CH1%>-X<%lgG1_up_hinge'->=>1&5<n>

The connector rule for standard pairing exemplifies the syntax as follows:

#standard pairing



- Rules are processed bottom down.
- **#** precedes comment lines that are not processed.
- Annotations can be added to the beginning of the rule divided by **:**. This annotation is shown in the AntibodyEditor details view.
- The rule notation separates domains by **-**.

- The domain acronyms refer to the domain library as follows:

Acronym	Chain	Domain
VH	heavy	variable
H	heavy	hinge
CH	heavy	constant
VL	kappa or lambda	variable
CL	kappa or lambda	constant
X	any	any

- Domains can be further specified by a text string to occur in the short name, set in <> brackets after the domain acronym, framed by ' .
 - % is used as a wildcard for 0-n characters.
 - ? is used as a wildcard for a single character.
- X is used as wildcard for any domain but usually applies e.g. to linkers.
- N- or C-terminal ends might be extended by * to indicate that the chain is allowed to be extended by e.g. modules like scFv but these * domains do not count when numbering the domains.
- Different chains are separated by , . Domains are continuously numbered across chains - just from left to right.
- The connection rule is given after the => symbol: The ordinal numbers of the domains to be connected by a Cys-Cys bridge are concatenated by & .
 - Per default it is assumed that such domains have only 1 free Cys left. Otherwise the ordinal number of the free Cys within a domain has to be specified after the domain number and separated by a # :
 - E.g. 3#2 specifies the 2nd free Cys of the 3rd domain.
- At the end of the syntax <1> or <2> indicate that the rule applies only if the given pattern is found once or twice, otherwise the rule is not applied. A <n> specifies that the pattern to be processed as often as it occurs.

The sequence of the rules define which rule will be used if two or more rules are matching. In this case, the first rule (lower sequence number) wins.

5.4 Protease library

The protease library is used to store protease descriptions which can be used to cut peptides.

The protease library (only 1st 4 lines) looks like:

protease_pk	protease_name	motif
1	Plasmin	^([ARNDCEQGHILKMPSTWVYX]*SCDK)THTCPPCP[A L][A]GGPSVLF[ARNDCEQGHILKMPSTWVYX]*\$
2	Papain	^([ARNDCEQGHILKMPSTWVYX]*SCDKTH)TCPCCPAPE[L A][L]GGPSVLF[ARNDCEQGHILKMPSTWVYX]*\$
3	MMP-3	^([ARNDCEQGHILKMPSTWVYX]*SCDKTHTCPPCPA)ELGGPSVLF[ARNDCEQGHILKMPSTWVYX]*\$
4	Cathepsin G	^([ARNDCEQGHILKMPSTWVYX]*SCDKTHTCPCCPAE)LLGGPSVLF[ARNDCEQGHILKMPSTWVYX]*\$

capture_group	appended_after	case_sensitivity	is_active
1	{null}	CASE_INSENSITIVE	1
1	{null}	CASE_INSENSITIVE	1
1	{null}	CASE_INSENSITIVE	1
1	{null}	CASE_INSENSITIVE	1

The following columns are included:

- **protease_pk**: used as internal primary key
- **protease_name**: name of the protease
- **motif**: peptide motif to cut
- **capture_group**: Defines the capture group that stays after cleavage
- **appended_after**: peptide sequence which will be added after the successful peptide cutting
- **case_sensitivity**: if the given motif is case sensitive or not. It is recommended to use CASE_INSENSITIVE
- **is_active**: only active proteases can be used in the HELM Antibody Editor

6 Contact

Please contact Stefan Klostermann (Stefan.Klostermann@Roche.com) in case of general and conceptional questions and feedback. Please contact Bernhard Schirm (schirm@quattro-research.com) for technical questions.