DIPLOMARBEIT

Titel der Diplomarbeit

„Vienna-PTM: Establishment of a server extending simulation capabilities of proteins by post-translational modifications“

Verfasser

Christian Margreitter

angestrebter akademischer Grad

Magister der Naturwissenschaften (Mag. rer. nat.)

Wien, 2012

Studienkennzahl lt. Studienblatt:	A 490
Studienrichtung lt. Studienblatt:	Diplomstudium Molekulare Biologie
Betreuer:	Dr. Bojan Zagrovic

Danksagung

Eine Diplomarbeit, und damit verbunden der Abschluss eines Studiums, ist nicht nur ein einschneidender Wendepunkt, sondern bietet außerdem die seltene Gelegenheit kurz innezuhalten und zurückzublicken auf eindrückliche, prägende Jahre. Dabei ist das Studium an einer Universität nicht nur eine berufliche Grundsatzentscheidung, vielmehr hat die schiere Vielfalt unterschiedlichster Eindrücke ganz automatisch einen beträchtlichen Einfluss auf die eigene Person. Dass diese Erfahrungen nicht nur positiver Natur sind, wird gerade im Rahmen einer Retrospektive gerne übersehen - man hat es ja schließlich trotzdem geschafft. Auf diesen Erfolg ist man stolz und freut sich über die zahlreichen Gratulationen - ehrlicher ist es aber sicherlich, dann auch auf all die zu verweisen, die ihn mit ihrem Beitrag erst möglich gemacht haben. Das möchte ich an dieser Stelle gerne etwas ausführlicher tun.

Den Anfang machen jene Personen, die in fachlicher und ganz generell wissenschaftlicher Hinsicht wegweisend für mich waren. Allen voran mein Diplomarbeitsbetreuer, Dr. **Bojan Zagrovic**, dessen Begeisterung für die wissenschaftliche Arbeit motivierend und ansteckend zugleich ist. Besonders hervorheben möchte ich neben dem außerordentlichen Vertrauensvorschuß, die beeindruckende Geduld und Sorgfalt, mit der er die vorliegende Arbeit korrekturgelesen hat. Herzlichen Dank auch an **Drazen Petrov**, der viel Arbeit und Herzblut in dieses Projekt gesteckt hat und dem, zusammen mit Prof. **Chris Oostenbrink**, **Melanie Grandits** und **Alexander Zech**, ein entscheidender Anteil an dessen Gelingen zukommt. An dieser Stelle möchte ich auch dem Rest unserer Laborgruppe danken, für kritische wie ermunternde Hinweise ebenso, wie für das ausgezeichnete Arbeitsklima und das (oft feuchtfröhliche) Zusammensein. Weiter zurückliegend, aber nicht weniger wertvoll ist für mich die Bekanntschaft mit Dr. **Wolfgang Reiter**, der mir durch sein Engagement nicht nur ausgezeichnete Einblicke in den Wissenschaftsbetrieb, sondern auch eine Praktikumsstelle an der ETH Zürich ermöglicht hat (ein besonderer Dank auch an Frau **Monika Kijanska**, die mich dort unter ihre Fittiche genommen hat). Abschließend möchte ich noch Herrn Prof. **Othmar Steinhauser** erwähnen, der mir durch seinen einmaligen, herrausragenden Vortragsstil die computergestützte Simulation von Biomolekülen nahegebracht hat.

So wichtig die fachliche Unterstützung durch Mentoren auch ist, sie kann den Rückhalt den man durch Verwandte und Freunde erfährt nicht ersetzen. Oft sind es diese Menschen hinter dem Vorhang, die den Unterschied ausmachen zwischen Erfolg und Niederlage. Ich darf mich deshalb ganz herzlich bei meinen Eltern, **Hubert** und **Judith Margreitter** bedanken, auf deren Rückendeckung ich immer uneingeschränkt zählen kann. Es ist euch nicht nur gelungen mich immer wieder zu ermutigen und aufzurichten, ihr habt mir auch ermöglicht, mich als Vollzeitstudent ganz auf das Studium zu konzentrieren. Dasselbe gilt natürlich für meinen Bruder **Georg**, der ebenfalls immer für mich da ist und nicht zögert, seine Hilfe anzubieten. Für all das und so viel mehr meinen innigsten Dank! Danke auch dir, **Sophie**, für deine zahllosen Aufmunterungen und hilfreichen Vorschläge. Mit dir an meiner Seite scheint mir nichts unmöglich. Meinem erweiterten Verwandtenkreis gebührt spezieller Dank dafür, dass sie mir mit Zuwendungen finanzieller und kulinarischer Natur das Leben versüßt haben - hervorzuheben sind in dieser Hinsicht meine Omas **Herlinde** und **Maria**. Abschließend ein herzliches Dankeschön an alle Freunde und Kollegen, mit denen ich in den letzten Jahren soviel Spaß hatte und auch manchen Erfolg feiern durfte (Liste keineswegs vollständig): **Alex**, **Beate**, **Bella**, **Christoph**, **Clemens**, **Flo**, **Franka**, **Kathi**, **Markus**, **Michael**, **Sandra** *et al.*

Danke!

Korrelation $\neq$ Kausalität

Contents

List of Figures

List of Tables

CHAPTER 1

Preface

1.1 Abstract

Post-translational modifications (PTMs) of proteins have, over the last decades, been extensively investigated from a number of different aspects. They are involved in a manifold of critical processes in the cell, including signaling, regulation and localization control. PTMs make fast and predominantly reversible amino acid alteration possible, are often inter-dependent and build sometimes networks on their own. The impact on the physical-chemical properties of affected residues is often significant and potentially affecting the overall properties of the whole protein. However, *in silico* simulations of PTMs have been strongly neglected, especially considering their biological relevance. With increasing numbers of observed types and occurrences of PTMs, it seems therefore timely and important to include them in classical mechanical force fields used for biomolecular simulations. This work presents **Vienna-PTM**, a server designed to provide both a workflow for introducing post-translational modifications in protein PDB files as well as parameters for these modified amino acids. Thereby, the arsenal of possible building blocks is enriched from 20 to over 200 distinct residues, including common modifications such as phosphorylation, acetylation and methylation as well as a number of less widely used ones. All modifications are available for GROMOS force fields ffG45a3 and ffG54a7.

1.2 Zusammenfassung

Post-translationale Modifikationen (PTMs) wurden während der letzten Jahrzehnte ausgiebig mit verschiedensten Methoden erforscht. Sie sind Bestandteil einer Vielzahl wichtiger Zellprozesse, einschließlich verschiedener Signalweiterleitungs-, Regulations- und Lokalisierungsvorgänge. PTMs ermöglichen die vorwiegend reversible Veränderung von Aminosäuren, sind oft abhängig voneinander und bilden unter Umständen ganze Netzwerke. Ihre Wirkung auf die physikalisch-chemischen Eigenschaften betroffener Reste ist oft signifikant und kann sogar Parameter des gesamten Proteins beeinflussen. Im Hinblick auf ihre hohe biologische Relevanz werden *in silico* Simulationen von PTMs jedoch nach wie vor stark vernachlässigt. Es scheint deshalb an der Zeit, PTMs in klassischen, mechanischen Kraftfeldsimulationen zu berücksichtigen. Die vorliegende Arbeit präsentiert **Vienna-PTM**, ein Server, der sowohl die Möglichkeiten für die Einführung von post-translationalen Modifikationen in PDB Dateien, als auch die notwendigen Parameter für die modifizierten Aminosäuren bietet. Dadurch wird die Bandbreite der zur Verfügung stehenden Proteinbausteine von 20 auf über 200 erweitert, wobei sowohl die häufigsten Modifikationen wie Phosphorylierungen, Acetylierungen und Methylierungen als auch weniger genutzte unterstützt werden. Alle Modifikationen sind für die GROMOS Kraftfelder ffG45a3 und ffG54a7 verfügbar.

1.3 Introduction

Amino acids, the building blocks of proteins, are often exposed to chemical alterations, either concomitantly with peptide formation (co-translationally) or afterwards (post-translationally). The latter, referred to as *post-translational modifications* or PTMs, play a number of key roles in a variety of cell processes. They are e.g. widely used for the regulation of enzyme activity, but are also involved in signaling cascades, cell-cycle control as well as regulation of translation and transcription (see also section 3). Of the 20 natural amino acids (also called *canonical*), 15 can be modified, thus creating a vast source of proteome diversification [81]. This high importance of PTMs is further emphasized by the fact that currently more than 200 different types of post-translational modifications have been described [81, 82].

The growing field of *in silico* simulation of biomolecules, chiefly proteins and nucleic acids, is gaining more and more importance in the last years. The two main advantages of molecular modeling / molecular dynamics techniques are the atomic resolution, which allows to elucidate mechanical details of the considered molecular processes with unmatched level of details on the one hand and the precise calculation of a manifold of characteristics, which are barely accessible otherwise, on the other. In that sense, computer-aided description and simulation of biological macromolecules can be seen as using a microscope with atomic spatial and femtosecond temporal resolution. In this respect, it is important to have in mind the limitations arising from various simplifications (see section 4.2) which are inherent in classical force field approaches. However, taking into account that the power of these computational methods scales directly with growing computer capabilities, one might expect their importance to increase in the future.

Considering the impact of post-translational modifications on the one hand and the potential inherit in molecular dynamics on the other, it seems quite surprising that there is almost no wide-range support for PTMs available so far. To the best of our knowledge, except for the parameterization of several specific PTMs (e.g. lysine and arginine methylation and lysine acetylation [30]), there is currently only one project aimed at the description of a larger set of PTMs - 27 in total, developed by Khoury *et al.* for the AMBER force field [37] (unpublished results). The main challenge in extending a building block database in this sense is the establishment of meaningful parameters, taking into account both the chemical nature of the modified amino acids and the general parameterization philosophy of the force field considered. Moreover, the introduction of "unusual" values should be avoided or at least be justifyable. Besides that, simulations including PTMs are further complicated in their application by the need for a modified initial protein structure file, which is often not available from databases and thus has to be generated. This can be time-consuming, since physical atom positions for each particular PDB file and modification have to be calculated, while also carefully matching specifications in the force field parameter files.

In order to tackle the difficulties mentioned above, Vienna-PTM - a dedicated webserver, which is able to introduce post-translational modifications into PDB files and serves as the source of required parameters for simulation - has been established. This diploma thesis concerns the workflow's usage and describes the internal structure and the set of available building blocks in detail. Moreover, it contains all technical specifications and information necessary to maintain or even extend the workflow. Finally, a short elaboration about force fields and post-translational modifications in general and GROMACS in particular sharpens the scientific and didactic aspects of this work.

Vienna-PTM availability: `http://coil.msp.univie.ac.at`

1.4 Abbreviations

1.4.1 *In silico* simulation

MM	**m**olecular **m**odeling
MD	**m**olecular **d**ynamics
GROMACS	**G**roningen **m**achine for **c**hemical **s**imulations [8]
GROMOS	**Gro**ningen **mo**lecular **s**imulation
AMBER	**a**ssisted **m**odel **b**uilding for **e**nergy **r**efinement
PDB	**p**rotein **d**ata**b**ank or the corresponding file format [63]
RTP	**r**esidue **t**o**p**ology
ITP	**i**nclude **t**o**p**ology
HDB	**h**ydrogen **d**ata**b**ase
TI	**t**hermodynamic **i**ntegration
TDB	**t**erminal **d**ata**b**ase
LJ-interaction	**L**ennard-**J**ones interaction

1.4.2 Computational

PHP	**P**HP: **h**ypertext **p**reprocessor
UTF-8	UCS **t**ransformation **f**ormat (see below)
UCS	**u**niversal **c**haracter **s**et
CSS	**c**ascading **s**tyle **s**heet
HTML	**h**yper**t**ext **m**arkup **l**anguage
JS	**j**ava**s**cript
UML	**u**nified **m**odeling **l**anguage
SSH	**s**ecure **sh**ell
MySQL	My **s**tructured **q**uery **l**anguage
HTACCESS	**h**yper **t**ext **access**

1.4.3 Biological

PTM	**p**ost-**t**ranslational **m**odification
mRNA	**m**essenger **r**ibo**n**ucleic **a**cid
DNA	**d**eoxyribo**n**ucleic **a**cid
HAT	**h**istone **a**cetyl**t**ransferase
HDAC	**h**istone **d**e**ac**etyltransferase
acetyl-CoA	**acetyl-co**enzyme **A**
H3K56	*means:* **h**istone 3 lysine residue 56
MAPK	**m**itogen-**a**ctivated **p**rotein **k**inase
HMT	**h**istone **m**ethyl **t**ransferase
NME	**N**-terminal **m**ethionine **e**xcision
ROS	**r**eactive **o**xygen **s**pecies
APP	**a**myloid **p**recursor **p**rotein
CDO	**c**ysteine **dio**xygenase
MPO	heme protein myeloperoxidase
PAPS	3’-**p**hospho**a**denosine-5’-**p**hospho**s**ulfate

1.4.4 Miscellaneous

pH	negative decimal logarithm of the hydrogen ion activity
pK_a	negative decimal logarithm of the acid dissociation constant
H-bond	**h**ydrogen-bond
NOE	**n**uclear-**o**verhauser-**e**ffect
RCSB	**R**esearch **C**ollaboratory for **S**tructural **B**ioinformatics
wwPDB	**w**orld**w**ide**PDB** (see above)

CHAPTER 2

Workflow manual

In this section, the workflow to generate PDB files containing residues carrying post-translational modifications is described from user's point of view. From starting a job by selecting appropriate options, via download of resultant files, through to installation of parameter packages, all interfaces are illustrated and explicated in detail. The whole procedure can be subdivided into the generation of a modified PDB file and the installation of the corresponding parameter files. The necessary input for each particular job specified by the user consists of the initial PDB file[6.2.1], additional settings such as the force field type and the selection of the required modifications. The final product is in any case the modified or minimized file as well as a zipped archive containing also all intermediates, the log file and (eventually) the generated topology files. All required force field parameters have been tested (see subsection 4.2.6) and repeatedly checked. They can be downloaded from our page as zipped archives, for installation details see description below.

2.1 Generation of modified PDB file

The first step is the alteration of a PDB input file in a way such that one or more canonical residues are replaced by the modified ones[1]. The header region of the PDB file is ignored and all non-selected side-chains remain unaffected. Only those which are supposed to be changed are tested for intrinsic validity and completeness to guarantee successful alteration afterwards. New residues are added directly in the ATOM section of the file, while numbers and names of all atoms are updated as required. The residue labels of the extended alphabet are unique and are all of size three characters to be consistent with the PDB file format definition[6.2.1]. In case of bugs or problems, one can use the board available on the web page to ask for support. It is also advised to read the comments[2.3] below.

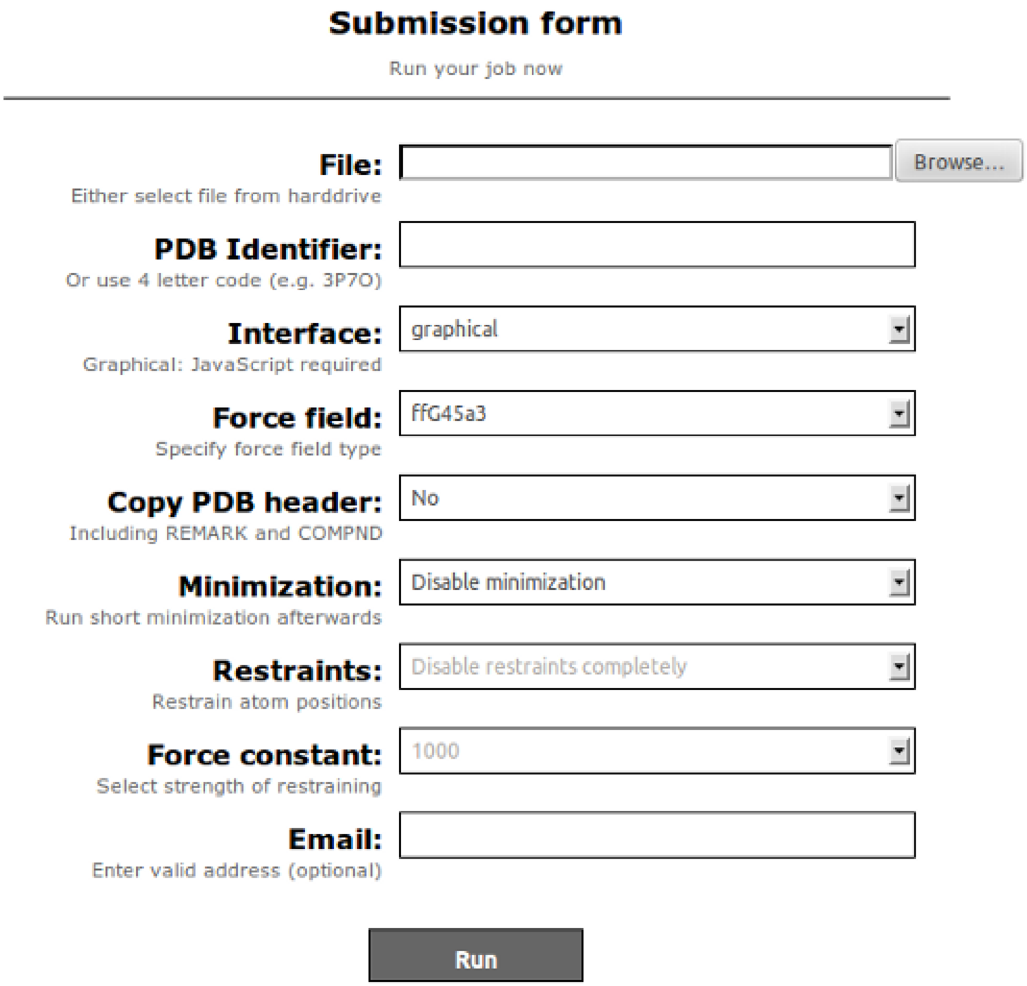
2.1.1 Upload of PDB file

The submission form for a new job is found on the frontend's start page (see picture 2.1) or under HOME in the main homepage menu respectively. It enables the following settings:

- **Upload of the PDB file:** Either by selecting it from the hard drive or by specifying a four-letter PDB code. In the latter case, the appropriate file is retrieved automatically from `www.pdb.org`. A local copy is stored on the server and included in the produced archive. The download and subsequent parsing may take a few seconds.
- **Interface:** The default is the graphical interface, which requires enabled JavaScript but offers additional information (i.e. polarity) about protein residues and a better overview in return. The text-based one is operational on virtually any system, independent of its configuration. Both implementations offer the same spectrum of modifications.

[1] For details see [4.1.2.2].

- **Force field:** Defines the force field, which will be used afterwards both on the server as well as on the user's machine. It is crucial to ensure, that this option is consistent with the installed package[2.2] since it will affect both the way atoms are attached as well as subsequent minimization.
- **Restraints:** Enables position restraints for the whole molecule except the modified residues.
- **Copying of the original PDB header:** If enabled, this setting will lead to a complete duplication of all header statements from the initial PDB file (the title will be changed though). See also the comments section[2.3] below, if problems are encountered regarding this option.
- **Email:** Optional setting. If an email-address is stated, the user will get a notification when the job is completed including download and deletion links.
- **Minimization:** If enabled, a short minimization is run afterwards to improve the geometry of the molecule. Although modifications are added initially with pre-minimized coordinates, it is recommended to either use the server's minimization or to run one locally before setting up simulations[4.2.5]. To run one's own minimization protocol, either the modified or the minimized file can be used as input.



Submission form

Run your job now

File: Either select file from harddrive	Browse...
PDB Identifier: Or use 4 letter code (e.g. 3P7O)	
Interface: Graphical: JavaScript required	graphical
Force field: Specify force field type	ffG45a3
Copy PDB header: Including REMARK and COMPND	No
Minimization: Run short minimization afterwards	Disable minimization
Restraints: Restrain atom positions	Disable restraints completely
Force constant: Select strength of restraining	1000
Email: Enter valid address (optional)	

Figure 2.1: Snapshot of the submission form menu on the start page including all available options.

2.1.2 Selection of modifications

The protein file is parsed chain-by-chain and the residues are displayed either by using the text-based interface or the graphical one. The number of available modifications is the same, amino acids are written in standard one- or three-letter code respectively. Selection of modifications can be carried out by moving the cursor onto one of the residues or by choosing from the drop-down list. The graphical menu shows all available modifications with the corresponding abbreviations on top of the residue (see table 2.1 for a complete list).

For some residues, more than one entry in the dropdown menu is available for a type of modification. This indicates one of the following:

- Different charges - modification: Each phosphorylation is supported both with minus one and minus two total charge, allowing the user to select the correct one, which is used afterwards in simulation. They differ also in their protonation states and are marked **(-1)** and **(-2)** respectively.
- Different charges - initial residue: Some amino acids come with a set of charge values because of their pH-dependence even if the modification itself does not. In such cases, the charge is stated the same way as above. For example, **(0)** indicates a neutral residue and **(-1)**, **(+1)** *et cetera* charged versions respectively.
- Multiplicity: Some modifications can be applied multiple times to one residue (e.g. methylation). The number of attached groups is given as **(#x)**, for example **(2x)** or **(3x)**.
- Multiple attachment points: Some amino acids can be modified at various sites. Tryptophan for example can be hydroxylated at five different positions, leading to the introduction of **1'**, **2'** *et cetera* postfixes to determine the exact location.
- Stereochemical versions: If residues differ only in one improper dihedral[4.2.3.4], they are marked with **(R)** and **(S)** to distinguish both conformations.
- Double methylation of arginine: The two methyl groups can be attached either in a symmetric **(s)** or asymmetric **(a)** way.

Those variations give rise to the total number of modifications in a group (see table 2.1). They may also occur combined.

Color code Polarity and charge state of residues are encoded in the color of the particular circle (graphical interface).

- Red (#): negatively charged (D, E)
- Blue (#): positively charged (K, R, H)
- Yellow (#): polar (N, Y, Q, S, T)
- Green (#): non-polar (V, A, I, L, C, M, G, F, P, W)

Modifications are also colored similarly following the approximate effect they would have on either the polarity or charge state of the amino acid. Figure 2.2 shows, how residue entries in the underlying PDB are manipulated according to the selection in the menu. The user starts the calculation and modification after a click on “Process” and is subsequently forwarded to a processing page until the job completes and the final result page is displayed. The required time for a job depends on the number of selected modifications, the size of the input file and whether minimization is enabled or not.

Modifications			
Ace	**acetylation:** Lys (1 group in total)	Hyd	**hydroxylations:** Tyr, Lys, Asp, Asn, Cys, Phe, Trp, Val, Leu, Pro (23 groups in total)
All	**allysination:** Lys (1 group in total)	Met	**methylations:** Arg, Lys, Glu, Asp, Gln, Asn, Cys, His (20 groups in total)
Bro	**bromination:** Trp (1 group in total)	N-t	**N-terminal modifications:** all (9 groups in total)
Cbm	**carbamylation:** Cys, Lys (2 groups in total)	Nit	**nitrosylation:** Tyr, Cys, Trp (3 groups in total)
C-t	**C-terminal modifications:** all (2 groups in total)	N-g	**N-glycosylation:** Asn (1 group in total)
Car	**carboxylation:** Glu, Lys (4 groups in total)	Oxi	**oxidation:** His, Thr, Pro, Met, Cys (6 groups in total)
Chl	**chlorination:** Tyr (1 group in total)	Pho	**phosphorylation:** Thr, Tyr, Asp, Ser, Lys, Arg, His (16 groups in total)
Deh	**dehydration:** Thr, Ser (2 groups in total)	Sul	**sulfation:** Cys, Met, Tyr (3 groups in total)
Nor	**isomerization (norleucine):** Met, Leu (1 group in total)	Cit	**citrulline:** Arg (1 group in total)
Kyn	**kynurenine:** Trp (3 groups in total)	Gsa	**glutamic semialdehyde:** Pro, Arg (1 group in total)

Table 2.1: Complete register of abbreviations used for the graphical interface: The colored balls are placed on top of the related residues. The amino acids, to which at least one of the members of a distinct group of modifications is applicable, are enlisted afterwards. Finally, the number of subgroups of modifications[6.1.2], represented by a particular colored ball, is stated[2].

[2] Meaning the number of distinct modifications, which share the same symbol.

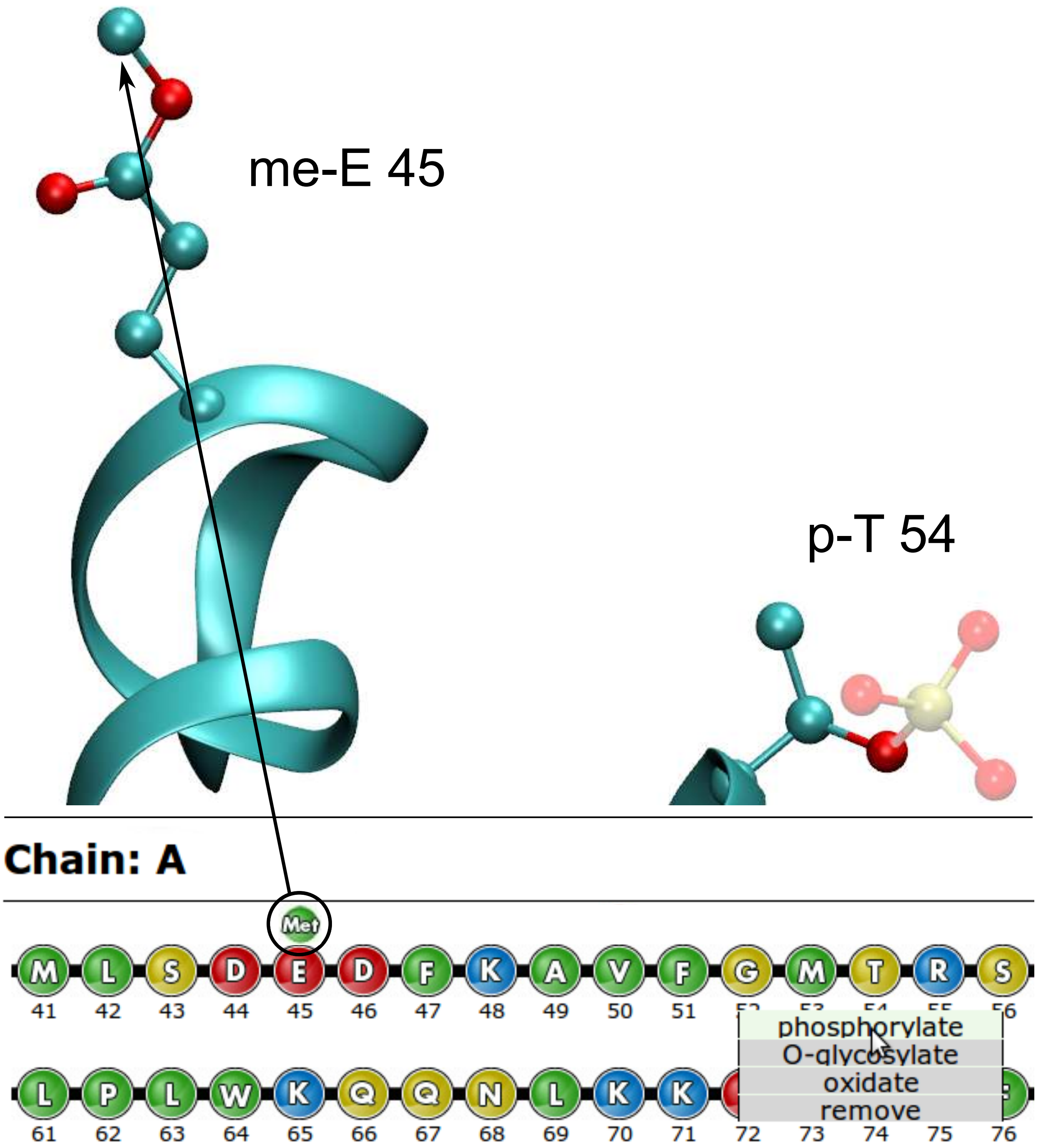


Figure 2.2: Illustration of the graphical selection menu (below) and the effect on the PDB file (top). Atoms are added in a pre-minimized way and can be arranged in a proper orientation by subsequent minimization. Unpublished results, [44].

2.1.3 Download of files

After job completion (or exceeding the time limit), the user is redirected to the results page. At the top, the logfile is loaded and parsed to monitor progress. If the last statement is “Job done”, a modified PDB file has been generated - even if minimization has not been successful or the initial validity check had failed. The next entry lists the modifications applied at the atomic level followed by the amino acid sequence[3] generated. The protein is also displayed in a Jmol frame (figure 2.3), followed by the download links (figure 2.4) and their description. The latter are also sent to the user by email, in case an address has been given. Finally a deletion link is stated, which will completely remove all job related data from our server.

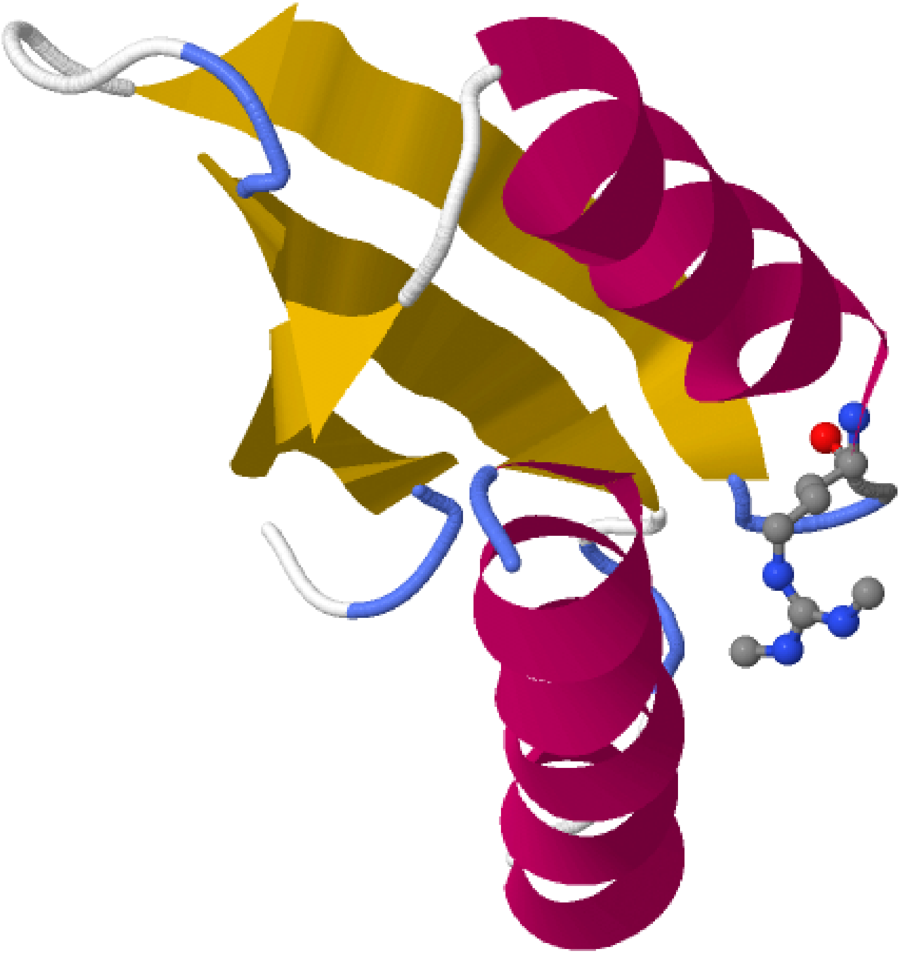
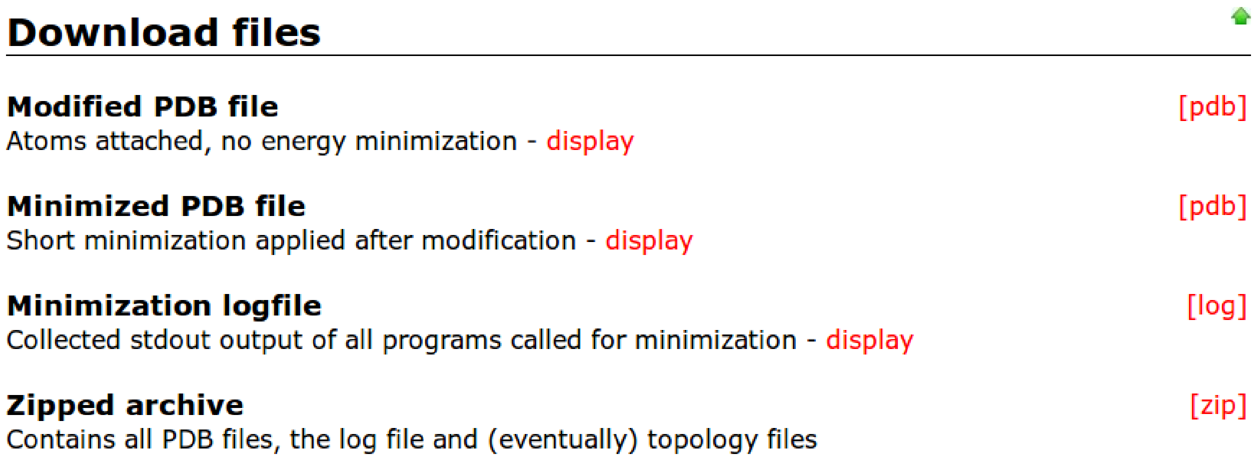


Figure 2.3: Jmol illustration of a modified and minimized protein based on file `3ZZP.pdb` taken from the result page. The canonical residues are represented by main-chain secondary structure (α-helices, β-sheets and turns), while the modified ones are in atomic detail. In this case, an arginine has been (asymmetrically) double-methylated. Zooming in or out can be achieved by using the mouse wheel and rotation by pulling the molecule with the left mouse button.

[3] Non-canonical residues, including the modified ones, are marked red.



Download files

Modified PDB file [pdb]
Atoms attached, no energy minimization - display

Minimized PDB file [pdb]
Short minimization applied after modification - display

Minimization logfile [log]
Collected stdout output of all programs called for minimization - display

Zipped archive [zip]
Contains all PDB files, the log file and (eventually) topology files

Figure 2.4: The PDB files can either be downloaded or opened in the browser (use "display"). All files are protected by a passphrase, to avoid unauthorized access. To download all related files including intermediates, logfiles and topology files at once, a zipped archive is additionally provided.

2.2 Installation of extended parameter files

To use the modified PDB files in combination with GROMACS, it is necessary to extend the library of available residues / building blocks which is located in the RTP file[6.3.4]. Moreover, it is also required to provide residue abbreviations and macromolecule type they belong to (in this case: protein) separately[6.3.5]. Additionally, if N- and C-terminal modifications are applied, the corresponding files have to be altered too[6.3.2]. In principle, there are two ways to accomplish this: either by replacing the original files or by usage of GROMACS inbuilt overrule ability. There are different package versions available for download, since the folder structure used in GROMACS has changed from 4.0.x to 4.5.x. Moreover, some file formats are different. Thus it is absolutely crucial to download the correct one.

2.2.1 Permanent installation

It is not recommended to replace the original GROMACS files, but if modifications are used frequently, it may be an option. Depending on your GROMACS installation, the force field input files are located in different folders[4]. Download the correct package from our server and install it either in `/PATH/gromacs/top/` (GROMACS version below 4.0.x) or `/PATH/gromacs/top/gromos45a3.ff/`. Make sure that you got the right version for your GROMACS.

2.2.2 Project-based installation

GROMACS also supports an overrule ability, whereby it first searches for parameter files in the current directory. Only if this fails, it loads the ones in the default directory. Therefore, it is sufficient to either copy folder `gromos45a3.ff` to the desired directory or to do the same with the files directly in case of a lower GROMACS version. Mind the varying filenames for different GROMACS versions. For example, if folder `testrun` contains the minimized file `min_3ZZP.pdb` (which has been downloaded from the server), package `ppackage_ffG45a3_GV3.zip` has to be unzipped in the same directory. The subsequent call of e.g. `pdb2gmx` is the very same as it was without the modified files.

[4] Common: `/usr/local/gromacs/share/gromacs/top/`

2.3 Comments

Since the PDB format changed over time, there is a big spectrum of possible statements for a file - some include details about the underlying experimental methods, others originate simply from the demand to include as much information as possible. Unfortunately, this also leads to files, which are corrupted in all ways imaginable, altough the main information - the coordinates - are valid. Therefore, the server ignores statements in some cases or only extracts parts which are crucial for modification. The same holds true for GROMACS, which enforces adjustments in some cases.

- Terminal modifications require minimization since those are applied by `pdb2gmx` (see chapter 4.1.2.4).
- ANISOU: Aniosotropic temperature factors are ignored.
- Protein chains and minimization: GROMACS ignores chain identifiers since it recognizes separated molecules directly. Therefore, the minimized PDB, which is a reconversion of a GROMACS topology file, does not contain chain identifiers either.
- Rotamers / alternative positions: The server always selects the first one and deletes the other. In case you want to use the second one, delete the first from your PDB file.
- Inconsistent atom names: There are cases where GROMACS renames specific atom names automatically during topology generation. This is usually not a problem, because these atoms do not occur in the canonical amino acids. However, they appear in some modifications. Thus the standard naming procedure is skipped for these residues$^{4.2.3.1}$.
- Copying of the PDB header: Since header entries are not changed during modification, it is recommended to use this option with caution.
- To enable broad usability and to ease up following simulations, modified residues are directly placed in the `ATOM` statements block.

CHAPTER **3**

Post-translational modifications

3.1 Proteins

Proteins are biological macromolecules which participate in almost all processes in living cells. They are normally composed of twenty (so-called canonical) *amino acids*. These small building blocks are covalently linked one after the other to form a linear chain of a length up to typically several hundred amino acids, in a process called *translation*. The principle of reconstructing complex biochemical compounds from simpler elements thus allows using the same machinery for generation of a vast variety of products, each of which is synthesized to fulfill a clearly defined function.

An outstanding subgroup of proteins are *enzymes*, which are necessary for many biochemical reactions in cells. They need to be accurately regulated, both in terms of efficiency and selectivity in order to maintain a healthy state in cells. This fine-tuning can be achieved by manipulating protein abundance (by controlling transcription, translation and degradation rates) or by switching proteins from active to inactive states and *vice versa*. The latter is often facilitated by introducing covalent changes of protein side-chains known as *post-translational modifications* (PTMs).

The change in the overall chemical structure of a protein introduced by a post-translational modification is typically rather small. For example, methylation of a distinct lysine residue in a long peptide chain corresponds to the addition of four atoms to a molecule consisting of thousands. However, the effect on the structure and dynamics caused by this tiny alteration can be tremendous. It can affect activity, selectivity, binding, localization and even stability. Therefore, information about sites and functional consequences of PTMs for a given protein are an integral part of its description.

3.1.1 Protein synthesis *in vivo*

Protein synthesis in cells is facilitated by large complexes, consisting of various regulatory proteins and ribosomes, the actual sites of synthesis. Ribosomes are ribozymes[1] and use an mRNA molecule as a template to generate a peptide chain. It stands to reason that the efficiency of translation of a particular mRNA is an excellent target for regulation of protein abundance. This can be realized through post-translational modifications of components of the translational machinery, e.g. in response to extracellular signaling.

[1] The enzymatic activity is actually performed by nucleic acids. As for ribosomes, RNA is responsible for peptidyl transferase activity.

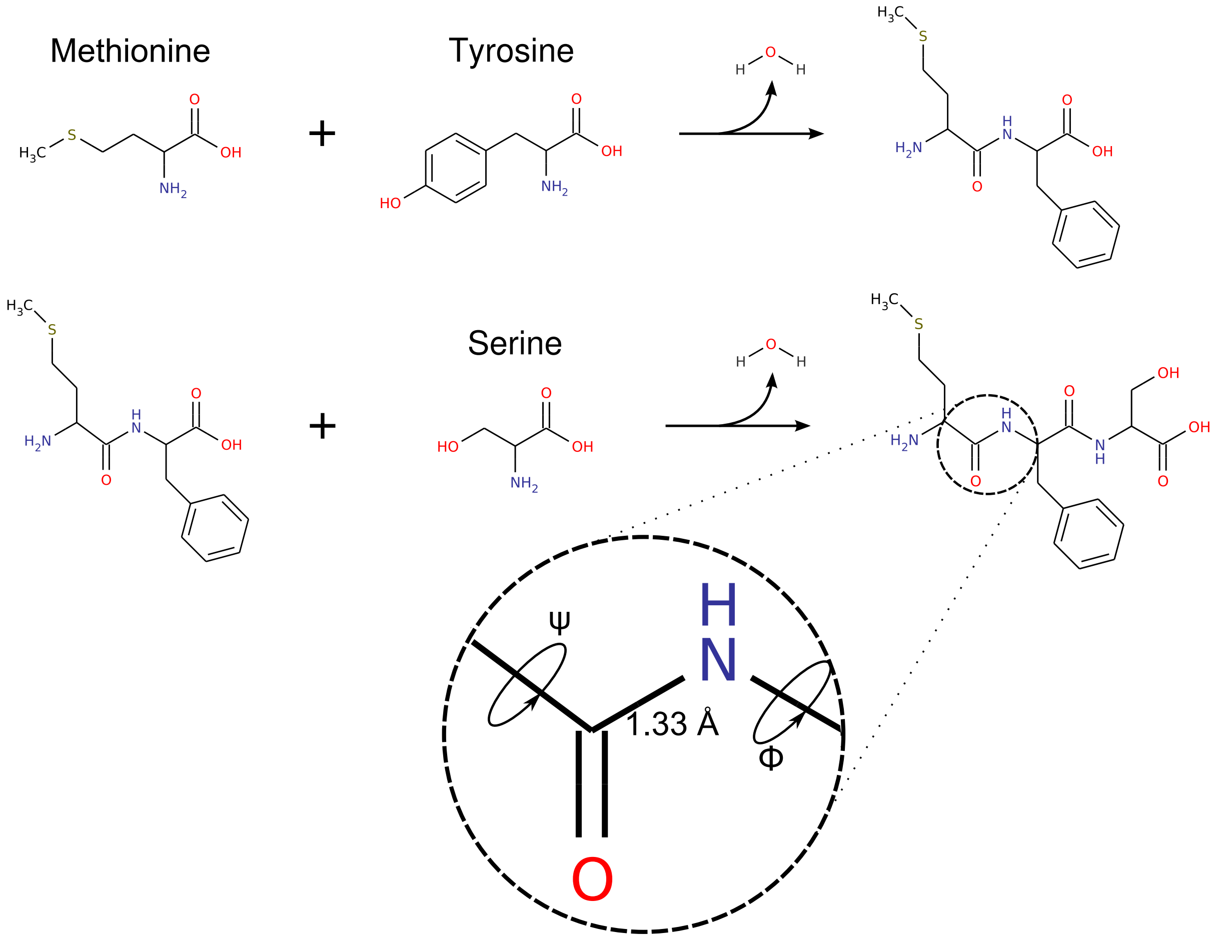


Figure 3.1: Illustration of peptide bond formation between methionine[2], tyrosine and serine catalyzed by *peptidyl transferase* releasing one water molecule per bond. The peptide bond itself is shorter than a normal C-N bond, due to its partial double bond character. The latter is also a part of the reason why rotation around this bond is restricted. Therefore, only two of the three backbone bonds of a residue can rotate significantly: C_α-C (Ψ) and N-C_α (Φ). Certain combinations of these angles are characteristic for distinct secondary structure elements. All residues tend strongly to have their peptide bond in the *trans* conformation, except proline for which the distribution is roughly even due to its special structure.

3.1.2 Amino acids and post-translational modifications

The building blocks of proteins are 19 amino and 1 imino acids. These monomers vary strongly in size, polarity and charge of their side-chains but are, in the context of proteins, all connected to one another by the same covalent backbone bonds. This range of physical-chemical properties renders it possible that the resulting macromolecules carry out all the different tasks that are required inside a cell. However, it is a significant advantage to use the same components over and over again, since it reduces complexity and allows recycling. Therefore, different strategies have arisen to minimize the coding effort (DNA) and the required production machinery while at the same time enable generation of different products necessary to build up complex organisms. Post-translational modifications are a powerful way to support this, since they allow (reversible) chemical alteration of particular residues after protein synthesis.

The spectrum of protein functions is limited by the range of chemical functionalities of their amino acid side-chains. Thus, to extend this range, there are cases where a chemical reaction is not facilitated by the protein itself, but by associated moities such as coordinated metal ions or bound RNAs. However, another way to enrich the functional potential of proteins without adding new amino acids to the alphabet are direct chemical alterations of amino acids through *post-translation modifications* (PTMs). Moreover, since the primary structure of a protein is permanent, reversible PTMs provide a mechanism well-suited for regulatory purposes.

Importance: A PTM is a covalent chemical alteration of amino acids. In fact, 15 residues are capable of being modified in one or several ways [81]. They offer a vast source of proteome diversification and are crucial for a wide range of cell processes. It is thought that about 5% of the human genome encodes for proteins which are involved in introducing, removing or reading of post-translational modifications. There are more than 200 different PTMs known to occur in living cells, with some being quite common and others less so. PTMs can be divided in two groups: enzyme-dependent and spontaneous (e.g. oxidation). Most PTMs are reversible and thus widely used for different kinds of signaling or (transient) activation.

Particularly common post-translational modifications are phosphorylation, acetylation, methylation, oxidation and glycosylation. Their relevance in (molecular) biology and medicine is elaborated in detail in the following pages.

Remarks: All modifications described in this chapter are supported by our server. For implementation details in a distinct force field see chapter 5. The diagrams representing examples for each particular group of modification use the standard color code for the atoms[3]. Abbreviations used to describe charge state, stereochemical version *et cetera* are listed in detail in subsection 2.1.2. A list of all available modifications and their corresponding internal groups is given in the appendix, subsection 9.2.2.

[2] In prokaryotes, a special formylated methionine is used at the N-terminus.

[3] Red ... oxygen, light blue ... carbon, dark blue ... nitrogen, beige ... phosphorus, yellow ... sulfur.

3.2 Acetylation

Supported: lysine[4].

Acetylation is a chemical reaction, which transfers an acetyl group -$COCH_3$ to a molecule. In biochemistry, acetylation is facilitated by acetyltransferases (also: transacetylases) and was first discovered at histone tails [4]. However, nowadays it is known that acetylation is also used for the regulation of non-histone proteins. Most of these targets are transcription factors, but the substrate range includes also cytoskeletal proteins, chaperones and nuclear import factors [27]. Only lysine residues are capable of being acetylated at their terminal end: the addition of the negatively charged acetyl group leads, in combination with the positively charged ϵ-amino group, to a modified residue with neutral overall charge. Besides that, acetylation of lysines can also extend the half life of proteins by blocking ubiquitinylation [81].

Histone acetylation: As described above, acetylation of lysine negates the positive charge thus lowering its interaction potential with the negatively charged DNA backbone. This is thought to reduce the packing degree of the chromatin structure, leading to an increased transcription rate in the affected chromatin region. Those histone acetylations occur at the histone tail, but in some cases - as for example the well studied acetylation of K56 in histone H3 in *Saccharomyces cerevisiae* [58] - lysines which are located in the core region are also affected. For H3K56, it has been shown that the particularly required acetylation state, depending on the cell cycle, is inevitable for proper histone function. The acetylation / deacetylation of this residue indicates DNA damage and is stabilizing replication forks [84]. The enzymes responsible for the right equilibrium of acetylations on histones are called *histone acetyl transferases* (HAT) and *histone deacetyltransferases* (HDAC). HATs use Acetyl-Coenzyme A (acetyl-CoA) as the source compound to facilitate the transfer.

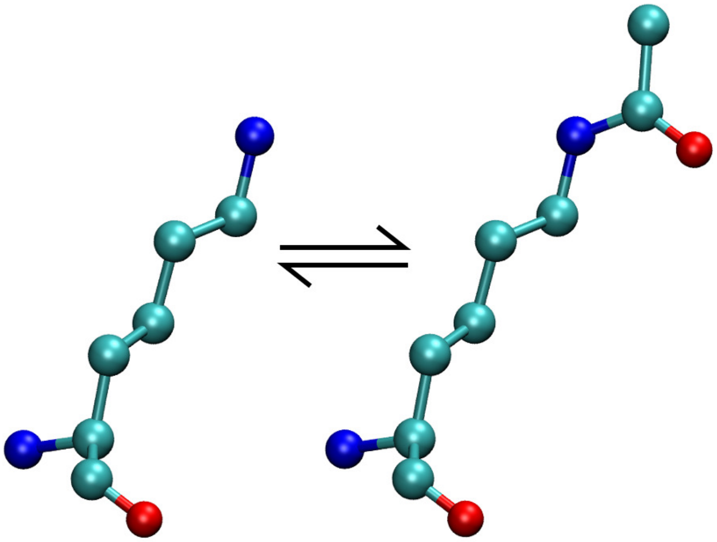


Figure 3.2: Acetylation of lysine: The positive charge contribution of the terminal amino group is neutralized by the addition of the acetyl group. Histones have N-terminal chains at which various modifications in different patterns can appear depending e.g. on the cell cycle stage, receptor driven adaptations and the type of tissue. This modification is specifically recognized by proteins containing a so called *bromodomain* [85]. This is one way to recruit specific transcription factors where they are needed at the right time.

[4] See also N-terminal acetylation.

Crosstalk: Acetylations are not only important cellular markers on their own, but they are also mutually dependent on other post-translational modifications [85]. Together, they build up a complex regulatory system both on histone tails and non-histon proteins [38]. Because of that and the assessed broad applicability, acetylation is now considered as an integral part of the cell's PTM arsenal.

3.3 Carboxylation / Carbamylation

Supported: carboxylated lysine and glutamic acid, carbamylated lysine, arginine and cysteine.

The term carboxylation refers to the addition of a carboxylic acid group -COOH to a chemical compound. In living cells, glutamic acid residues are the main target and the one discovered first was one in prothrombin, a factor in the blood clotting pathway [76]. This modification plays an important role in hemostasis, bone calcification (see below) and, in organisms such as *Conus textile*, ion channel regulation [7]. Moreover, lysine residues can also be altered in this way at their terminal amino group, shifting the total charge of the resulting amino acid towards neutral. Carboxylated lysines seem to be generated without a certain catalyzing enzyme [39]. Most often, they are required for building hydrogen bonds or for metal ion coordination. Furthermore, they appear (rarely) at catalytic sites where they are responsible for proton transfer: a known case is *OXA-10-β-lactamase*, where this modification is required for proper function.

Carboxylation and blood clotting: Carboxylation of Glu residues is facilitated most frequently by the protein γ-glutamyl carboxylase. This enzyme oxidizes Vitamin K and simultaneously attaches a -COOH group to the particular side-chain. The main fraction of targets belongs to proteins involved in blood clotting. Those are activated by γ-carboxylation: γ-carboxyglutamic acid renders binding to calcium and phospholipids possible [76]. Vitamin-K dependent carboxylation occurs at sites containing the so called *γ-carboxylation recognition sequences*, which are bound with high specificity by the carboxylase. The target is only released after multiple carboxylations in this region, leading to area-wide carboxylation [7]. The clotting factors require 10 to 12 carboxylated glutamic acids (all within a range of about 40 amino acids), which bind calcium ions as bidentate chelators [81] leading to a change in conformation in which the affected proteins tend to aggregate in complexes. Furthermore, they activate others which are nearby in order to initiate a subsequent chain reaction.

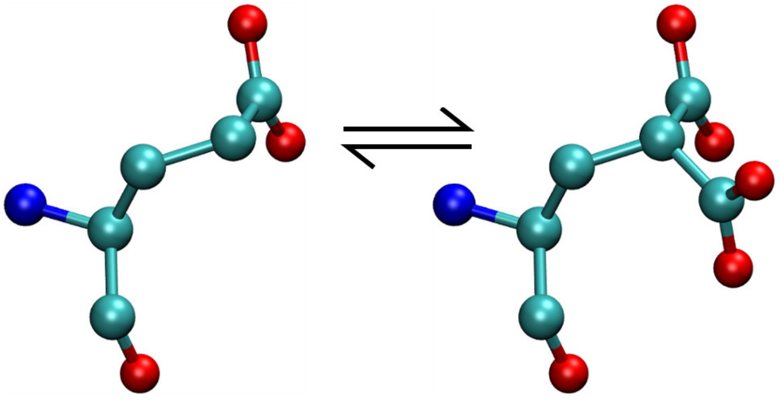


Figure 3.3: Shows carboxylation of glutamic acid as it occurs during activation of blood clotting factors. Deficiency of Vitamin K or defects in the gene encoding for γ-glutamyl carboxylase result (among others) in the loss of proper coagulation. The exact underlying mechanism and the order of binding events necessary is still under investigation [31].

Carbamylation: The addition of a -CONH group to lysine, arginine, cysteine and the N-terminus is known to occur accidentally in protein-urea solutions [40]. Besides that, it has been reported to be a key modification for the functionality of some proteins like class D β-lactamases [28]. Additionally, carbamylated arginine (also called homocitrulline) is supposed to be a key player in developing autoimmune diseases, for example, at induction of arthritis [50].

3.4 C-terminal modification

Supported: amidation (all), methylation (C, L, K).

Amidation: The amidation of C-terminal amino acids (*alpha*-amidation) in proteins is common for peptide hormones: more than 50% of neural and endocrine proteins are altered in this way [19]. This modification is applied by a protein called *peptidylglycine α-amidating monooxygenase*, an enzyme exchanging an oxygen of the terminal carboxy-group with an amide in a two-step fashion [21]. This process requires ascorbate, oxygen and peptidylglycine[5] as the donor of the amino group and copper ions as cofactor. Although neutral amino acids are the most common target, amidation has been predicted to be possible for all canonical amino acids by cDNA analysis. It is thought, that this modification is necessary to prevent ionization of the carboxy-terminus in order to support binding to receptors by increasing hydrophobicity.

Methylation: The methylation of the C-terminal residue forms an ester with its carboxylic acid. Mainly proteins containing motif `-CaaX` are affected[6], where the three residues are removed and the cysteine is both methylated [25] and prenylated [86]. This mechanism e.g. is necessary for the regulation of stability of Ras and Rho proteins, which are proto-oncogenes and therefore intensively studied [16].

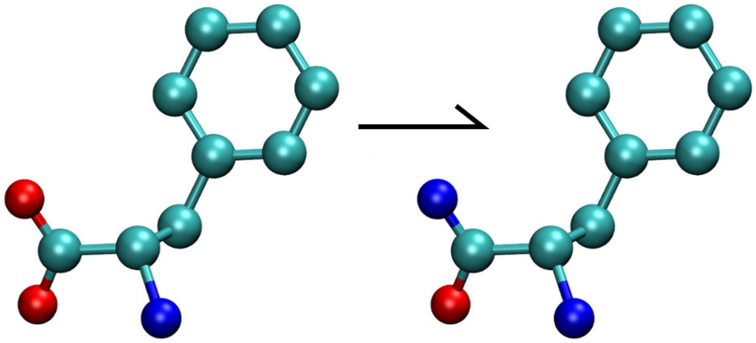


Figure 3.4: Illustrates amidation of a C-terminal phenylalanine residue. Replacement of one oxygen by an amino group leads to neutralization of the negative charge.

[5] Sequence: XGXX, where X stands for any residue.

[6] C ... cysteine, a ... aliphatic residue, X ... any residue.

3.5 Phosphorylation

Supported: serine, threonine, aspartate, tyrosine, histidine (1', 3'), arginine and lysine (all available both in -1 and -2 versions).

Phosphorylation is a PTM of high importance: It is e.g. used for regulation of binding events or to induce conformational changes in proteins [81]. Moreover, it is crucial for the initiation of ubiquitin dependent degradation of some proteins and the stabilization of others. The important tumor-suppressor protein p53 for example seems to be influenced in its stability by phosphorylation [6]. In general, the addition of the negatively charged -PO_3^{2-} group can introduce local changes in the protein's conformation, leading to the propagation of structural alterations. Moreover, many proteins contain several different sites where phosphorylation can occur, thus resulting in surface patterns which can be used gradually to tune their catalytic activity. The equilibrium of phosphorylation states is controlled by kinases - enzymes, which transfer a phospho-group onto a protein - and their counterparts, phosphatases [81].

The reversible nature of phosphorylation enables fast adaption to a changing environment and thus they are predestined for signaling purposes. For example, *Pseudomonas aeruginosa* has more than 130 proteins which are part of a two-component regulatory system [64] driven by phosphorylation events. These systems consist of a receptor with the ability for auto-phosphorylation on a histidine residue and a regulation part which is used as a signaling relay and can be phosphorylated on a defined aspartate. This induces response, often by directly facilitating DNA binding and transcription or by enzymatic events.

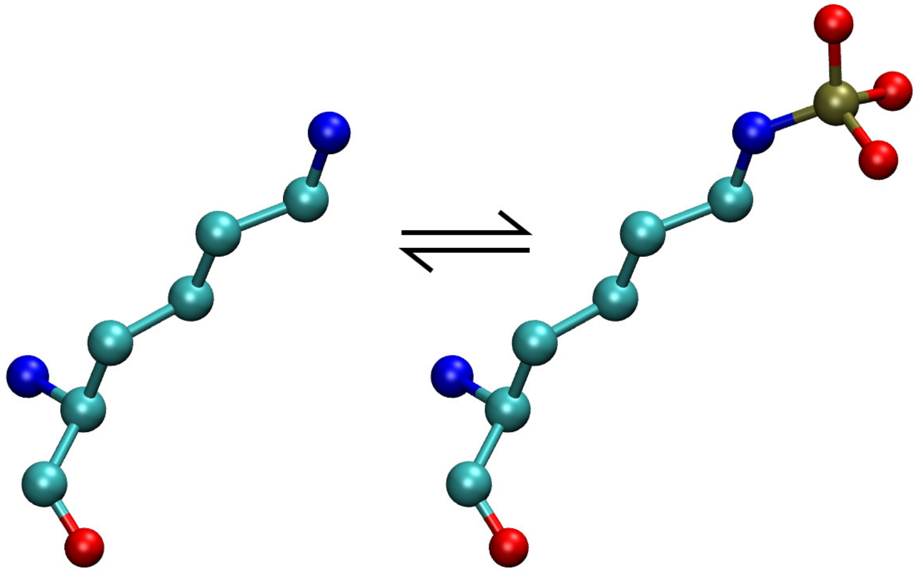


Figure 3.5: Phosphorylation of lysine: The addition of three oxygen atoms results in a strong shift of the residue's charge. Lysine, which is a positive residue in most environments, carries a negative charge after phosphorylation.

Other examples are *mitogen-activated protein kinases* (MAPK) cascades [70], which are transducing an incoming signal to the nucleus through a network of serine/threonine kinases in order to induce changes in the protein expression pattern. Some kinases are specific, others have multiple targets. While it appears, that yeast has several almost independent MAPK pathways side-by-side, there is more cross-talk observable in vertebrates. This allows to integrate different input signals (e.g. stress or growth signals) in order to show the appropriate response.

Distribution: p-Aspartate and p-histidine are only common in bacteria and fungi [33]. In mammalia, serine is by far the most common template for a phosphorylation (around 90%), followed by threonine (9%) and (far behind with 1%) tyrosine [42]. The wide applicability is also underlined by the huge repertoire of kinases and phosphatases: the human kinome[7] consists of over 500 members, most of them with multiple targets [43].

3.6 Hydroxylation

Supported: tyrosine, lysine, aspartic acid, arginine, cysteine, valine, leucine (3', 4'), proline (4', 5'), phenylalanine (2', 3', 4'), tryptophan (2', 4', 5', 6', 7').

Directed hydroxylations are mediated by protein hydroxylases. Those responsible for proline, lysine and asparagine are members of the family of non-heme Fe^{II} monooxygenases [81]. Two histidine and one aspartic acid side chains coordinate half of the iron coordination sites. Two more sites are filled by α-ketoglutarate and the last one by O_2. This leads, after O_2 cleavage, to a complex which is able to hydroxylate inactivated C-H bonds. The substrate amino acids are in general not electron rich or nucleophilic - the enzyme with its iron group renders attachment possible.

Hydroxylysine: Hydroxylysine was first discovered almost 100 years ago [80] and is a very important constituent of collagen. Its fraction in animals in all lysine residues depends strongly on the collagen type (type II two to four times more than type I or III [53]. It appears almost exclusively at the Y position of a triple helix (see below) and it is inevitable to maintain cross-links, both between and within chains. Hydroxylysine can also be a starting point for further modification i.e. glycosylation in collagen and other proteins.

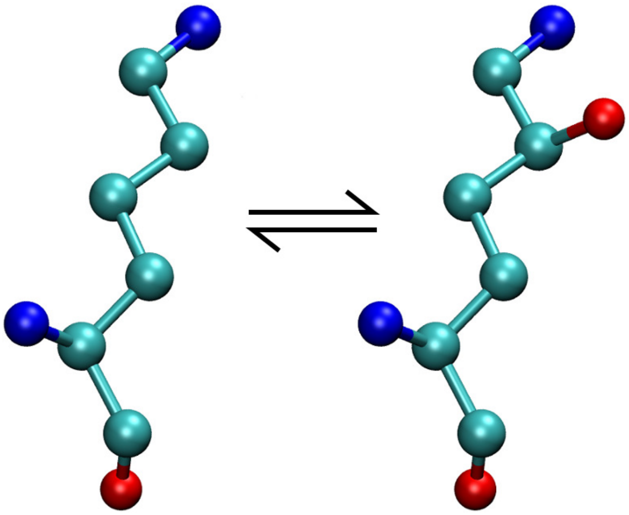


Figure 3.6: Illustrates hydroxylation of lysine, as it happens in collagen and a few other proteins. After hydroxylation, facilitated by the enzyme *lysylhydroxylase*, a subsequent glycosylation may follow. Hydroxylysine is also present in bacteria, where it is incorporated in the cell wall [72].

Hydroxyproline: Proline is by far the most frequent target for hydroxylation and 4'-hydroxylation of proline is the most common post-translational modification in humans all together [29] - constituting around 4% of all residues in animal proteins. This reaction is irreversible and requires Vitamin C as a co-factor - a lack of ascorbic acid leads to scurvy, a avitaminosis. The proteins, containing hydroxylated prolines cover a wide spectrum of functions - however, its role in collagen

[7] A superfamily of enzymes responsible for protein phosphorylation.

stability has been investigated the most thoroughly. The hydroxyl oxygen provides stereoelectronic effects, which increase collagen stability [9, 34]. Furthermore, each collagen chain contains motifs such as X-Y-glycine, where X is often proline and Y hydroxyproline, in order to stabilize this triple-helix. The possible impact of this modification on proteins other than collagen is versatile - ranging from changing conformations and binding interactions to being the initial step of a modification cascade. From a chemical point of view, the addition of a hydroxy-group is the introduction of a highly electronegative function, lowering the pK$_a$ value of the nitrogen from 10.8 to 9.68 and stabilizing the pyramidal form [20].

3.7 Methylation

Supported: arginine (1x [0,+1], 2x [s0,a0,s+1,a+1]), lysine (1x [0,+1], 2x [0,+1], 3x), aspartic acid, glutamic acid, asparagine, glutamine, cysteine, histidine (1'[0,+1], 3'[0,+1]).

Methyltransferases use *S-Adenosyl-methionine* as the donor for the $-CH_3$ group [81]. Most abundant targets are side-chains with a terminal nitrogen namely lysine and arginine.

N-Methylation: Methylations of lysine and arginine residues are among the methylations which have been studied most [59], because they are part of the histone code. This is fortified by the fact, that 7 of the first 36 residues of histone H3 can be methylated. The terminal amino group of lysine can be methylated once, twice or thrice - this makes highly defined regulation possible. On histone tails, the effects of methylation may be opposed by acetylation by means of activation of transcription. This states a wide range of regulation potential: a histone H3-dimer yields 14 methylation sites and 8 capable of acetylation. The enzymes responsible for methylation are *histone methyl transferases* (HMT), each one specific for one target residue. An example for the interdependency of these modifications is the methylation of H3K4 [77]. The methylation of H3K9 is blocked if the histone tail is already methylated at residue K4. This code is extended by additional modifications: H3K9 cannot be methylated, if H3S10 is phosphorylated or the residue is covered by acetylation. Together, this relationships between PTMs are considered as a complex regulatory network.

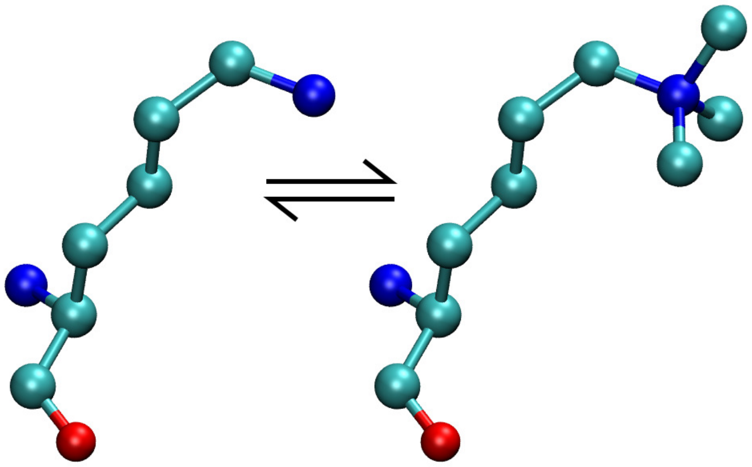


Figure 3.7: Picture shows trimethyllysine as it occurs at histone tails. Recruiting mechanism: each degree of lysine methylation, leads to binding to different proteins: for example, triple-methylated H3K9 is bound by the chromodomain of protein HP1 [36].

Double methylation of arginines [73], which is unique for eukaryotic cells, can be symmetric or asymmetric and is often related to RGG tripeptides. As for example, the localization of *RNA helicase A* is determined by the arginine methylation status at its RGG-rich *nuclear localization sequence.*

O-/C-/S-Methylation: Although less abundant, methylations of aspartic acid, glutamic acid and cysteine [81] are also known[3.4]. They can be used as markers: if aspartic acid is isomerized, which occurs spontaneously during cell aging, methylation of iso-aspartic acid by *L-isoaspartate-O-methyltransferase* is the first step in recovering aspartate residues [15]. Methylation of glutamic acid in membrane proteins is known [2] to play an important role in the ability of bacteria to follow nutrient gradients (chemotaxis). Again, addition of a methyl moiety serves as a (reversible) marker, although the consequences of (de-)methylation may be different from one species to another. In the process of repairing alkylated DNA, the protein MGMT restores the native base by cleaving *O^6-methylguanine.* One cysteine is methylated during this process and the protein is degraded afterwards, probably because the formed bond is quite stable [69].

3.8 N-terminal modification

Supported: acetylation (all), pyroglutamic acid (Q, E), formylation (M), pyruvic acid formation (S, C, V), methylation [0,+1] (all), dimethylation [0,+1] (all), trimethylation (all).

Compared to C-terminal modifications, the possible alterations of the N-terminus are much more diverse. An important one is the N-terminal methionine excision (NME), which removes the initial methionine in about 80% of yeast proteins [22] thus enriching the fraction of possible start residues [26]. Enzymes, which are specific for distinct terminal residues are therefore able do distinguish N-termini, which allows specification. Moreover, this irreversible and co-translational process plays an important role in regulating protein turn-over rates. The penultimate residues which become exposed after NME are often non-bulky ones like alanine, cysteine or valine. In general, proteins which are abundant, undergo NME more often than the rarer ones [45].

Acetylation: N^{α}-terminal acetylation is one of the most abundant modifications in eukaryotes. Most proteins in human and yeast (84% and 57%, respectively) are modified in this way: *N-acetyltransferases* transfer the acetyl group of cofactor acetyl-CoA to the α-amino group of proteins [5] and each of these enzymes has a defined subset of target residues. For some proteins, this reaction affects only a fraction of the total population. As for the function of this modification, it is thought that it may mark proteins for their subsequent localization in the cell[8] [22] and (partly) for degradation[9] [35, 48]. Moreover, it seems to be related to NME since proteins retaining their initial methionine are not acetylated [45].

Methylation: The proteins exposed to N-terminal mono-, di- or trimethylation vary in terms of function, but it appears that many of them are part of large complex structures, for example ribosomes, myofibrils, nucleosomes or flagella [75]. It has been supposed that the first few residues function as a recognition site for the methylation machinery. A well studied example is *regulator of chromosome condensation 1* (RCC1), the guanine nucleotide exchange factor for Ran-GTPase [32]. It is important for many processes that RCC1 generates Ran-GTP at precisely defined locations, e.g. in the case of mitotic spindle assembly and nuclear envelope formation. This is enabled by N-terminal methylation of RCC1 (stable throughout the cell cycle), because this modification is

[8] Proteins destined for the cytosol are often acetylated, those transferred to the endoplasmatic reticulum are not.

[9] Acetylation can work as a "degron" (degradation signal): modified N-termini are recognized by the E3 ligase *Doa10* and the protein is ubiquitinylated.

able to stabilize the colocalization with mitotic chromosomes and interphase chromatin.

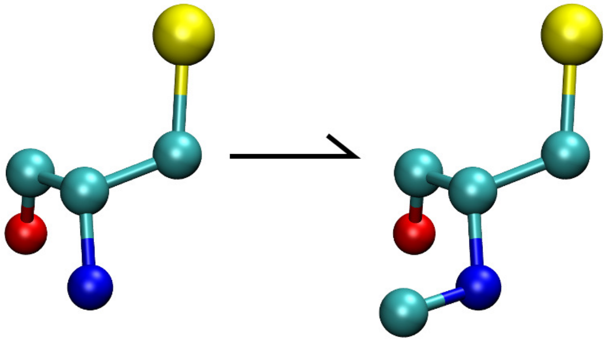


Figure 3.8: Illustration of N-terminal methylation of cysteine. The positive charge of the amino group is diminished by the addition of the methyl group.

Others: Besides acetylation and methylation, N-termini are, among other modifications, formylated and glutamic acid is transformed into pyroglutamic acid, a cyclic derivative. The latter reaction occurs quite often and can be induced both spontaneously or by an enzyme [83].

3.9 Oxidation

Supported: pyroglutamic acid, histidine, threonine, methionine, cysteine.

Oxidative modifications of proteins [10] are chemical reactions in cells with radical species, which often contain reactive nitrogen or oxygen, leading to non-functional protein derivatives. They are connected to aging as well as different diseases, especially neurodegenerative ones. Oxidation by radicals can be induced by electron leakage, metal-ion-dependent reactions and autoxidation of lipids and sugars [17]. The type of oxidation depends on the radical species. Radical formation may lead to cascade formation, but the primary free radical is most often the superoxide[10] radical $O^{2-}\cdot$. Itself, it is of low reactivity, but it can be the precursor for highly reactive radical species like peroxyl, alkoxyl and hydroxyl radicals. Moreover, it is thought, that cell growth is connected with the oxidative state, since various transcription factors including AP-1 and NF-κB are redox-controlled [17]. Sulfur-containing and aromatic amino acids are preferred targets of *reactive oxygen species* (ROS), tryptophan for example may form kynurenine.

Repair: The major effects on proteins are their (partial) denaturation and thus inactivation as well as subsequent aggregation. To avoid reaching dangerous thresholds, the main strategy of the cell is (specific) degradation. The oxidized protein's accessibility to proteases grows with each additional oxidation[11]. Most oxidations are irreversible [10], with the only exception being derivatives of methionine and cysteine, which are of a high susceptibility. Therefore, a repair machinery is needed to restore proper function.

[10] The main reaction producing this radical are electron leakages from mitochondria electron transport chains.

[11] Up to certain threshold, above which it starts to decrease.

Accumulation and diseases: If the balance between degradation by proteases and formation of oxidized proteins is disturbed in a way that protein derivatives start to accumulate, this leads to the creation of protein aggregates. Such agglomerates play an important role in Alzheimer's disease, muscular dystrophy, rheumatoid arthritis, amyotrophic lateral sclerosis *et cetera* [10]. For example it is known, that accumulations of Aβ peptide, a cleavage product of *amyloid precursor protein* (APP), and fibrils are the molecular basis for developing Alzheimer's disease [65]. Those extracellular plaques are thought to cause damage to synapses [46].

Controlled oxidation: There are cases, when oxidation is not the result of random reactions of radicals with susceptible functions in proteins but facilitated by an enzyme. For example, *cysteine dioxygenase* (CDO) [47] oxidizes cysteine residues in order to form cysteine sulfinic acid (see below). If this enzyme is not able to work properly, this may lead to autoimmune diseases and neurological disorders.

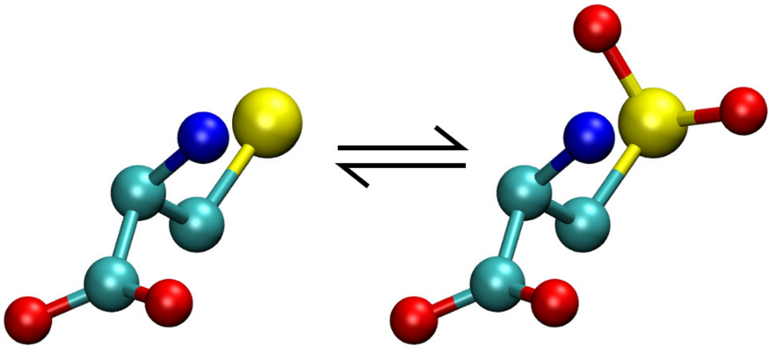


Figure 3.9: Shows oxidation of cysteine to form cysteine sulfinic acid. This reaction is catalyzed by the enzyme CDO.

3.10 Other modifications

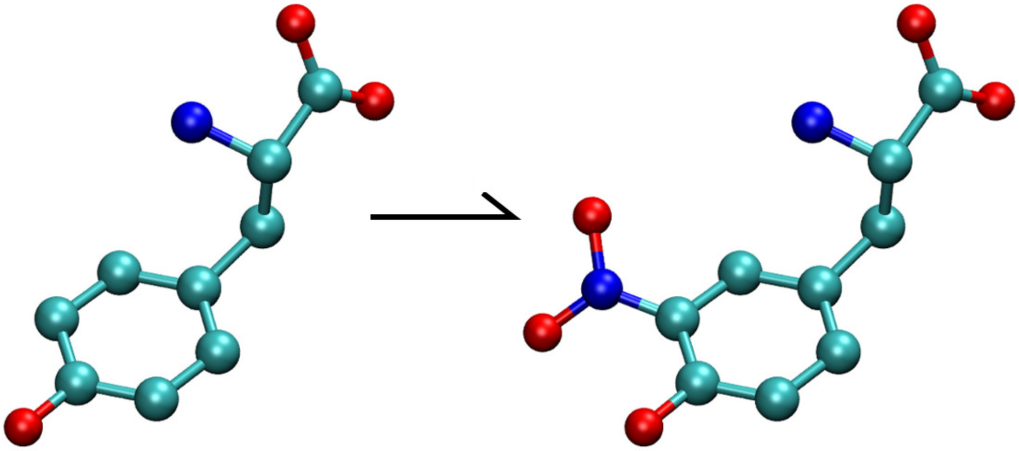


Figure 3.10: Nitrotyrosine is a marker for oxidative damage present in a lot of different diseases including atherosclerosis, sepsis *et cetera* [49]. The enzyme *heme protein myeloperoxidase* (MPO) generates inflammatory signal mediators such as nitrotyrosine, by producing peroxynitrite, a product of MPO-mediated oxidation of nitric oxide. Besides that, there are also other pathways for MPO to increase nitrotyrosine levels. This modification may be reversible: it is possible that cells use specific *denitrases* to recover the initial tyrosine residue [18].

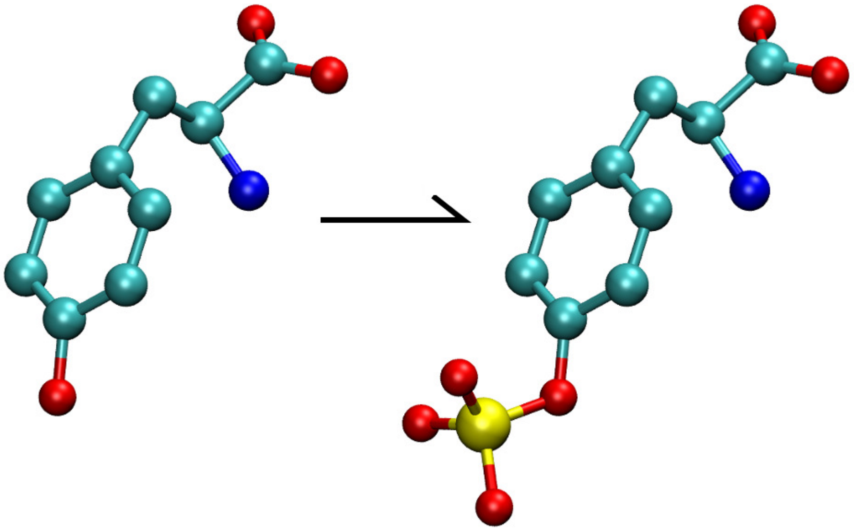


Figure 3.11: Sulfotyrosine is an abundant modification, occuring in most secreted and transmembrane proteins[12] [71]. A *tyrosylprotein sulfotransferase* transfers a sulfate group from *3-phosphoadenosine-5-phosphosulfate* (PAPS) to the oxygen at the aromatic ring in tyrosine [51]. Sulfation is important for inter-cell communication, growth and defense [71]. A biologically relevant example is leukocyte chemotaxis where tyrosine sulfation is necessary for the interaction between receptor and ligand. It is also connected to diseases, e.g. receptor CCR5 has to be sulfated in order to facilitate entrance of the HI virus.

[12] One percent of all tyrosines in eukaryotic cells is supposed to be sulfated. In prokaryotes and simple eukaryotes, this modification has not been observed.

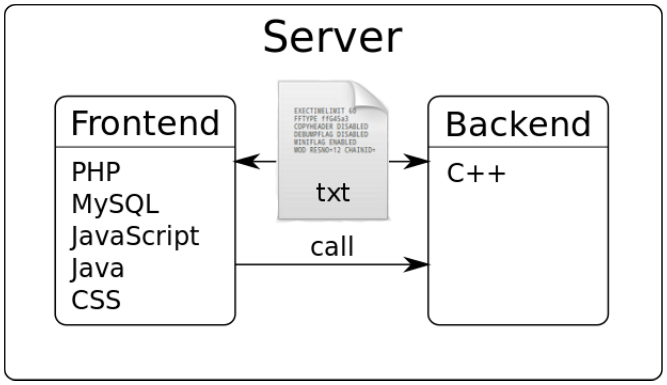
CHAPTER 4

Methods

In this chapter, all the techniques and components which have been used to implement the server are described. This also includes a description of the main scripts and their functions. Moreover, a short summary of the theoretical background of GROMOS force fields and GROMACS engine as well as an introduction to the key parameters for classical force fields are given. Furthermore, the development and refinement of these parameters for the supported post-translational modifications is discussed with a focus on the embedment in the server's framework. Finally, the technical server specifications are listed in detail.

4.1 Server structure

The steps of the workflow are distributed between two distinct parts: the frontend and the backend module. Both are executed on the same physical machine and communication is mainly facilitated by generation and propagation of instruction files (text files, UTF-8 encoded) except for the initial call of the backend. The crucial advantage of this segmentation is increased stability, since a failure in one module should not affect the operability of the other. Moreover, it makes possible extensions and monitoring easier and decouples the webserver virtually from the modification process itself[1]. Finally, (optimized) C++ allows a fast and efficient implementation compared to PHP, which is in turn the perfect choice for webserver scripting.



```
EXECTIMELIMIT 60
FFTYPE ffG45a3
COPYHEADER DISABLED
DEBUMPFLAG DISABLED
WINIFLAG ENABLED
MOD RESNO=12 CHAINID=
```

Figure 4.1: Illustration of server structure and underlying techniques. The backend does not access the database in order to simplify automated input in case of a large-scale approach[2]. All the information required is either stated in parameter files (plain text) or passed by the call. Therefore, a script generating proper instruction files for a set of jobs would be able to do multiple calls automatically. See also section 4.3.

[1] As long as the defined interfaces do not change, the modules can be treated independently.

4.1.1 Frontend specification

The frontend is basically the collection of scripts and database tables constituting the homepage. It includes the user interfaces which grant access to the server's functionality, job distribution to the backend and precise monitoring of job progression. Moreover, it includes a powerful administration suite to maintain the key features via webmasks.

4.1.1.1 Scripts (server-side)

The following scripts are written in PHP and facilitate job preparation and presentation of the results as well as general functions necessary. Each page is split into three parts: header[3], content and footer whereas only the middle one differs. Thus, changes made in one of these files are adopted instantaneously on all pages. Consistent design is enabled by global `CSS` definitions (mainly `mainlayout.css`). Templates have not been used.

Main scripts:

- `index.php`: Contains the job submission form[2.1] and (depending on panel settings) changelogs and news, which can be directly edited by authenticated administrators. The submission method of the form is `POST`, because of the amount of data expected.

- `display_residues_form.php`: Parses the specified input file. In case an identifier has been given, the requested file is retrieved from `www.pdb.org` in advance. If minimization is requested, an initial validity check is performed which is technically a `pdb2gmx` call followed by subsequent output analysis. In case this test fails, minimization is automatically disabled for this job and a warning is printed. This may happen due to the presence of non-canonical residues, corrupted input files or unsupported ligands. To overcome this problem, one can either fix the input file or run an energy minimization locally using the resultant `mod_XYZ.pdb` file. Details about the two available interfaces can be found in subsection 2.1.2. For the graphical one, a JavaScript code is embedded in the script. Selections of modifications are buffered in a hidden textfield[4] and passed to `preprocessing.php` by `POST`. The text-based interface in contrast has one drop-down menu for each residue and is therefore passing a lot more data. On the other hand side, it only requires standard HTML elements, which raises compatibility to a maximum. Finally, `display_residues_form.php` creates the folder structure for the new job and adds the concerning entries to the database (see table 9.5). Those also include the job identifier and the related passphrase[4.1.1.6].

- `(pre)processing.php`: Generates the instruction file[6.1.1] out of the passed data. Moreover, it calls the backend and checks the job's status every few seconds[5]. The call is made in the background to avoid delayed script response. If the job is completed (indicated by the phrases "Job done" or "Job failed" in the logfile), the user is redirected to the final result page. Redirection is facilitated by using the HTTP-header.

- `sresex.php`: Displays the logfiles, the (altered) protein sequence, a Jmol-based 3D representation of the protein and download and deletion links. All residues are listed grouped by the chain they belong to. Non-canonical ones are printed in red.

[2] Meaning many different modifications on one or multiple input PDB files.

[3] The header includes standard functionality required for all scripts such as the establishment of the database connection and session support.

[4] Deletion or replacement is also facilitated on this level.

[5] The page refreshing interval can also be set by the admin panel.

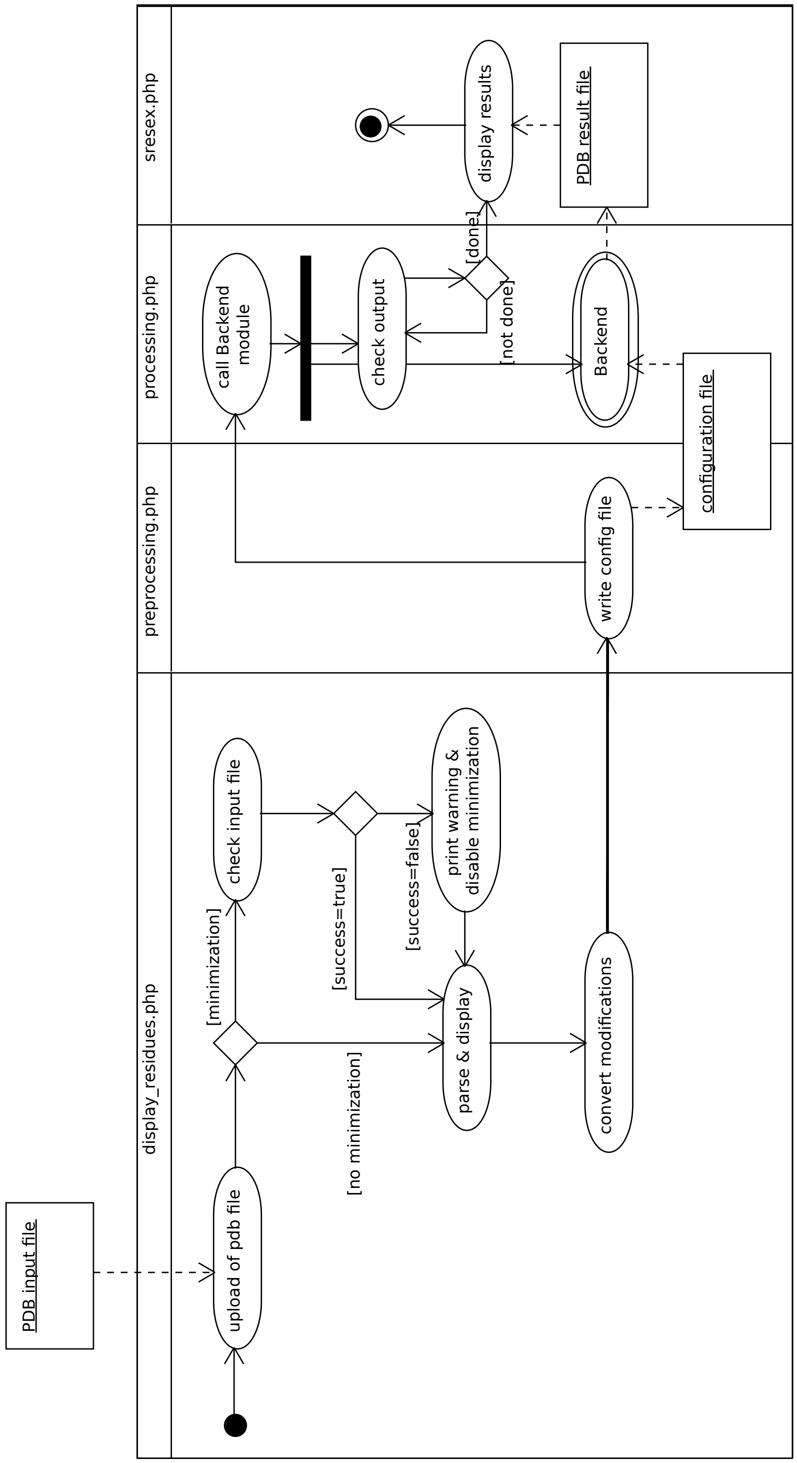


Figure 4.2: Frontend's activity diagram (UML) for a job. Note that there is also a check for minimizability in case terminal modifications are selected, because they require `pdb2gmx`. Database interaction is not shown.

- `downloads.php`: Provides download of parameter packages and miscellaneous documents as well as a maintenance feature, which allows administrators to manipulate the content directly. All information on downloads is saved in the corresponding database table `SITE_DOWNLOADS`[9.6].

- `about.php`: A short version of the manual section in this document. Most important specifications are listed. The content of this script can also be changed directly, however it has to be plain HTML. Script `literature.php` can be maintained the same way.

- `panel.php`: The administration panel is described in detail in section 4.1.1.7 below. It requires authentication by logging in prior to accepting any changes. As mentioned before, most pages can be changed after authentication directly: these scripts provide a change and / or deletion button on the bottom.

Because of security reasons, there is no possibility to change PHP scripts within the homepage framework. In order to achieve that, the server has to be accessed directly via `SSH`. If formats or parsers are changed, it is important to mind differences in `line brake` encoding between the modules.

4.1.1.2 Classes

All classes used in PHP scripts are listed in picture 4.3. Class `CFG_Object` is used as a container for the generation of instruction files, `residue` for parsing the input file (see above) and `stuff` for several duties including server-wide settings based on database entries. Moreover, `job` stores all the information on the current job and is used to update table `AUTH_USER_PERMISSIONS` (see 9.5). `downloads` is a stable interface for changing download database[9.6] entries by a web form. Finally, the file `header/basic_functions.php` contains widely-used standard procedures as string parsing or file access.

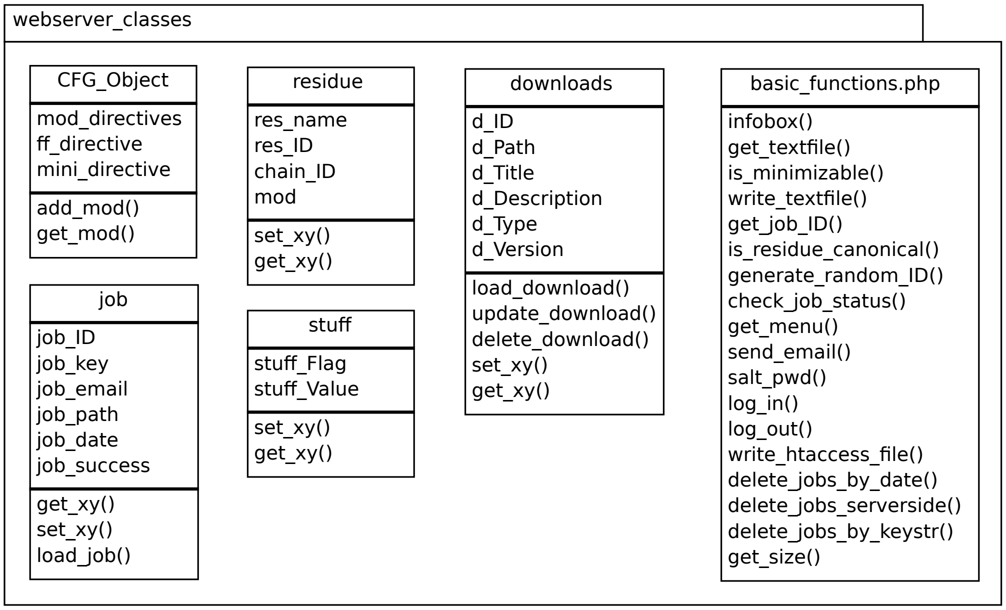


Figure 4.3: Shows all implemented frontend classes and relationships (UML [41] class diagram).

4.1.1.3 Miscellaneous homepage settings

For global settings, unique and fixed entries have been made in the table SITE_STUFF[9.7]. They are identified by their names and therefore order is not mandatory. Each entry consists of an identification phrase, a value (if necessary) and a flag (to enable or disable it).

Number	Name	Description
1	maintenance_site	Contains both the text of the maintenance note as well as the corresponding flag. If stuff_Flag is true, all jobs are blocked except for authenticated administrators.
2	default_res_per_line	The maximal number of residues in one table row (graphical menu).
3	default_text_res_per_line	The maximal number of residues in one table row (text menu).
4	refresh_time_processing	Defines the interval in which the two scripts preprocessing.php and processing.php check the progression of a particular job.
5	workflow_current_version	Specifies the version number used in the title.
6	auto_delete_jobs	Enables or disables automatic job deletion and defines the maximum time job data is stored.
7	minimization_available	Enables or disables minimization globally.
8	literature_text	Stores the HTML content of script literature.php.
9	about_text	Stores the HTML content of script about.php.
10	max_exec_time	Sets the maximum execution time (in seconds).

Table 4.1: Shows all entries available in version 1.1. New ones can be added in the same way, the only condition is to use unique identifiers. These entries are the only exception, where blank creation of the database tables is not sufficient to restore full functionality. However, it is recommended anyway to use a complete MySQL backup in case the server has to be reinitialized. For this purpose, *mysqldump* is an appropriate choice.

Typically, these flags are changed by the admin panel. To use them, it is sufficient to create an instance of class SITE_STUFF and load it:

Listing 4.1: Example for stuff call

```
// generate object $new_stuff
$new_stuff = new stuff();
/* load data from database:
first argument specifies which entry
second one is the object */
get_stuff( "identifier", $new_stuff );

// example usage
if( $new_stuff->get_Flag() == 1 ) // flag set?
  echo 'Value: ' . $new_stuff->get_Value();
```

4.1.1.4 Scripts (client-side)

The scripts executed on the client's side are actually the ones responsible for the graphical menu (embedded in display_residues_form.php) and the Jmol-Java plugin rendering the resultant PDB file on the final page of the workflow (sresex.php). To ensure maximum compatibility with any possible user setup, neither is crucial for proper workflow functionality.

The JavaScript code is an application of the jQuery based menu toolbox *Superfish*, version 1.4.8 [12], which integrates the *scriptaculous* library [24]. Its duty is to control the MOD statements[6.1.1] in the subsequently generated instruction file. Therefore, it requires functions to add, delete and search specific lines in the buffer text-area which are composed of the residue number, the chain identifier and the type of desired modification (see also description above).

The Jmol applet loads the modified or minimized PDB file of the current job and displays it. The call is modified by sresex.php to show the modified residues in atomic detail, while the others are represented by their secondary structure. The call is parsed by the Javascript library after object initialization.

Listing 4.2: Jmol applet call

```
jmolInitialize( "JmolFolder" );
jmolSetAppletColor( "#FFFFFF" );

jmolApplet( [800, 550], "load jobs/job_[NUMBER]/min_[FILENAME].pdb;
background [xFFFFFF];
set highresolution on;
set antialiasDisplay on;
select all; color structure;
cartoon; wireframe off;
set showHydrogens off; spacefill off;
select [MODIFIED]:; color CPK; spacefill 120;
wireframe 45; backbone off;" );
```

4.1.1.5 Job management

Each job has its unique identifier cur_job, which is a number generated by a MySQL entry insertion into table AUTH_USER_PERMISSIONS[9.5] and a related passphrase (jkey) which both are passed from one script to the next by usage of GET. On each page, the whole table entry is loaded to an instance of class job if the provided passphrase is correct. This allows one to run jobs simultaneously and simplifies subsequent access via hyperlinks (e.g. from email).

Note: cur_job should not be mistaken for the number which is part of the job's path. To render a direct input to the backend possible, this number is incremented by one not only for each new job started by the frontend, but also for each job solely executed by the backend. This is necessary to avoid clashes.

4.1.1.6 Security

As described above, access to job related files is only possible if jkey is correct. This passphrase is a string generated randomly and of a minimum length of twenty characters. It is also encoded via GET in the download links provided both in the email and the result page. Moreover, it is required to delete a job from the server: All files are removed completely - only the database entry specifying date, cur_job and success remain for the purposes of tracking statistics.

The server has a powerful framework for user administration, which is currently applied only to authenticate administrators. However, it could be easily extended to serve as a platform for resource management and individual usage monitoring. All users are enlisted in table AUTH_LOGIN[9.3] as well as their double-salted and hashed passwords. Once logged in successfully as either USER or ADMIN (both controlled by the panel, see below), PHP sessions enable further usage of functions restricted to users / administrators only.

All critical directories and files are protected by HTACCESS directives to avoid manipulation. Standard MySQL injections are prevented as far as possible.

4.1.1.7 Admin panel

The admin panel allows maintenance by setting a large spectrum of options.

- Miscellanous settings:
  - Activate / deactivate maintenance mode: If this option is on, only administrators are able to start new jobs. Other users will receive a message, which can be by administrators in the corresponding text area (HTML is allowed).
  - The number of residues per line for both the graphical and the text-based interface can be set.
  - (pre)processing.php refreshes the page every few seconds to check if the job is already completed. The interval length can be set.
  - Minimization can be generally disabled in order to reduce server load or for maintenance reasons. Again, this option has no effect on administrators.
  - The maximum execution time (in seconds) specifies how long a job is allowed to last. If this time is exceeded e.g. because the input molecule is too big to minimize it within time, the process is stopped.
  - The running version of Vienna-PTM stated in the header line can also be set.

  - All tables of the database are backed up each week. However, there is an option to do and retrieve a backup instantly.

- Options for user maintenance allow addition and deletion of users.

- Jobs can be deleted either manually (all or older than 1 day / 1 week / 1 month) or automatic deletion can be activated including specification of the maximum file perpetuation. Moreover, the current storage allocation caused by jobs is printed.

- File manipulation can be done for:

  - all `css` files used for the design.
  - each GROMOS force field: building blocks `[rtp]`, C- and N-terminal specifications `[tdb]`, hydrogen file `[hdb]` and residuetypes `[dat]`.
  - each force field: modification database file `[cfg]`, minimization file `[mdp]`, command files for minimization and debumping `[par]`.

- To set the available modifications (four-letter abbreviations) for each force field - residue combination, an interface can be used which is based on a PHP script and database table `POSSIBLE_MODS`$^{9.4}$ `[dbs]`. These selections are saved as serialized arrays in the corresponding table entries.

- To monitor the usage of the server in terms of (successful and failed) jobs, two graphs are provided (see picture 4.4 and 4.5).

- Finally, a visitor counter provided by `www.flagcounter.com` is used to show the number and origin of (unique) homepage accesses.

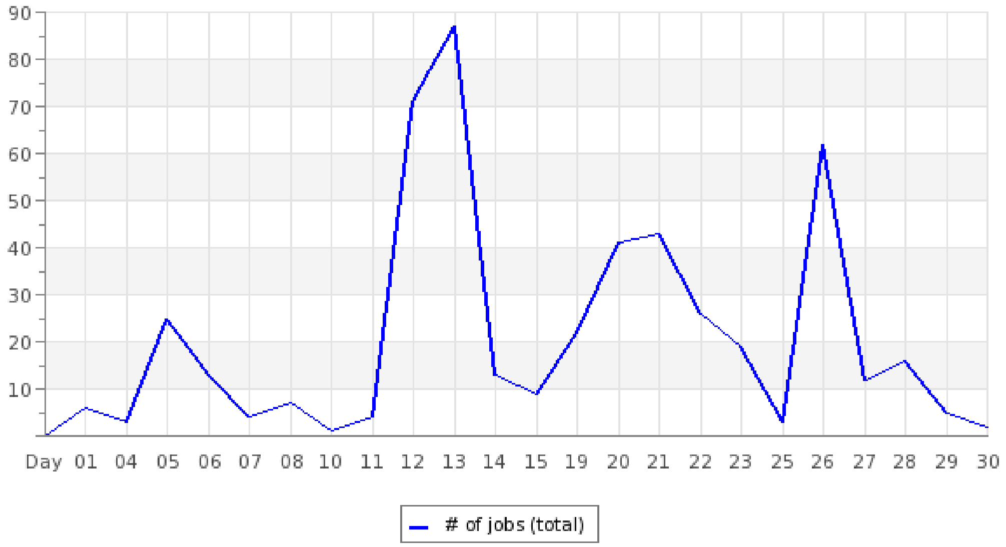


Figure 4.4: The server usage graphs show the job count per day for the current month. Does not take into account the final success state. Each job is counted independently from the user, meaning that a hundred jobs started by the same user are also visualized as a hundred jobs in this graph.

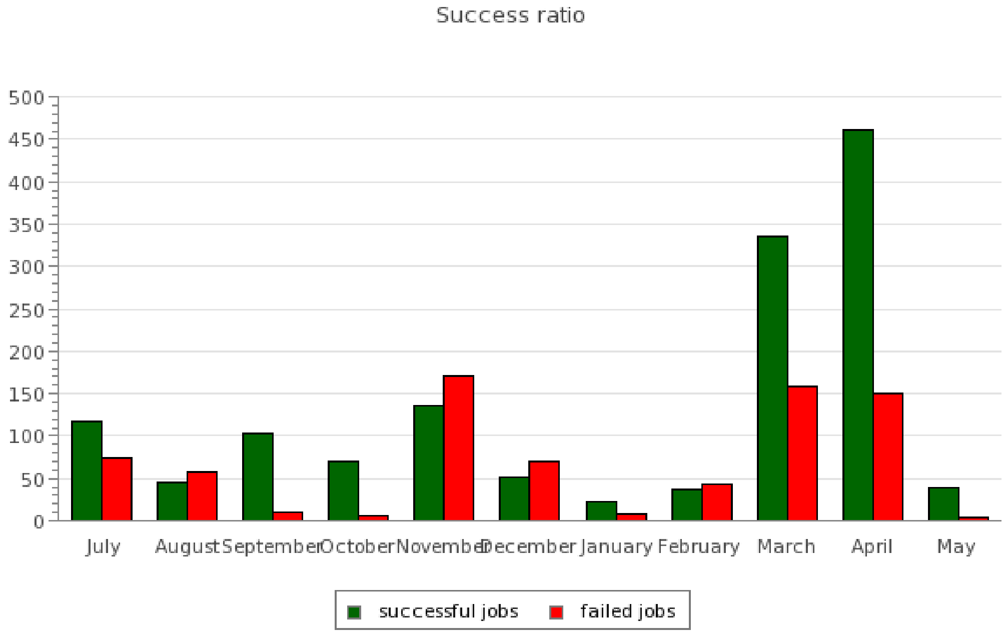


Figure 4.5: All jobs ran so far can also be itemized by months. The green bars correspond to successful jobs, while red bars refer to failed ones. If modification is successful, but minimization took too long or had to be disabled in advance, the job is counted as success. In contrast, jobs which have been started but remain unfinished, either because of a crash or lack of user input, are considered to have failed.

4.1.1.8 Board

In order to structure parameter development and to offer a feedback possibility for end users, a board [1] has been installed at `http://coil.msp.univie.ac.at/wbb`. Access to developer forums is restricted to specified user groups, while reading permissions are granted to everyone and posting is enabled after free registration.

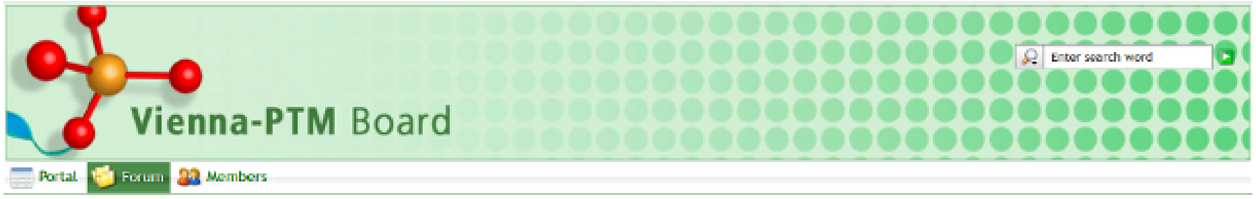


Figure 4.6: Header of support board.

4.1.2 Backend

While the frontend is required to transform the user input into a well-defined format[6.1.1] and to initiate and monitor subsequent processes, the backend is the actual site of modification and file manipulation. Except for the job's path, all information regarding a particular job is passed by the corresponding instruction file. The end product is, independent of the steps in between, a zipped archive including logfiles, the input PDB file, the modified and (in case) minimized output files and topology files.

The backend module is written in C++, following the object-oriented (class) paradigm[4.1.2.1]. The main object encapsulating all data structures and methods required to process a job, is an instance of class `PDB_Object`. It parses the input PDB file, the job instruction file and several database files (for details see activity diagram 4.8 and subsection 4.1.2.2). All requested files are loaded at once, reducing the input-output load to a minimum, while the demand for additional memory is justifiable. The content of these files is decomposed and buffered in appropriate data structures and (in case required) reassembled to enable write-out. Important file format definitions are set globally (e.g. the ranges of data blocks building an ATOM line in PDB files).

Listing 4.3: Core excerpt of `RESEX.cpp`

```cpp
int main()
{
// Initialization
PDB_Object obj;

// Tasks
obj.load_file( input );
obj.analyze();
obj.apply_modifications();
obj.write_out();
obj.minimize();
obj.zip_files();

return 0;
}
```

Note that only one parameter has to be parsed and provided initially: the `input` path of the PDB file (seen from the backends' directory). For all subsequent operations this relative path (see subsection 6.1.3) is added as prefix to the file names.

4.1.2.1 Classes

While some classes are only used as data containers providing methods to read, write and analyze certain file formats, others are more sophisticated and cover functional aspects in the workflow. All functionality, except basic functions for string manipulation, data type conversion *et cetera*, is encapsulated in distinct objects. This is not true for occasions, where additional functions or different strategies which had arisen during development, would have led to big alterations in the initial class design: in this cases, global parameters have been used to circumvent strict encapsulation.

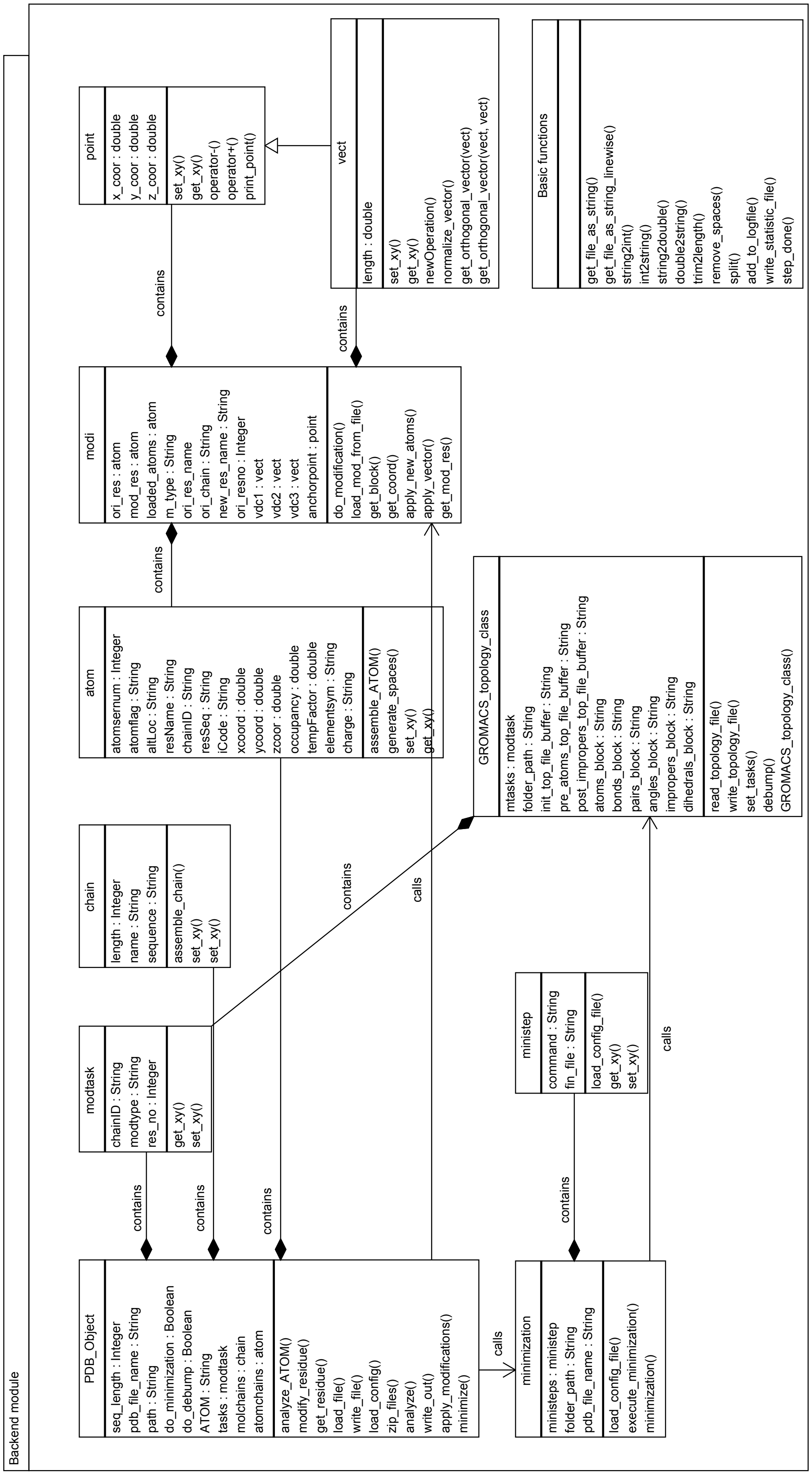


Figure 4.7: Shows all implemented backend classes and relationships (UML class diagram) implemented in the backend.

4.1.2.2 Backend workflow path

Since each job is called independently and the working directory can be specified freely in the execution command, it is possible to automatize execution of multiple jobs by a script. The only restriction is the fact that parameter files are not duplicated. Therefore, it is not possible to use different setups (in terms of e.g. building blocks) without changing centralized parameterfiles at least transiently.

The workflow path for the backend is as follows:

1. The PDB file$^{6.2.1}$, specified by a command line parameter in the backend call done by the frontend is loaded: `./backend -i jobs/job_[NUMBER]/[FILENAME].pdb`. Note that all input files are expected to be in the very same folder and all output files will be deposited there as well. The ATOM statements of the PDB file are stored as standard strings in a vector and further analyzed to be finally transcribed into instances of class `atom`.
2. The corresponding instruction file$^{6.1.1}$, whose name has to be `[FILENAME].pdb.cfg`, is loaded. Information about the used force field version, minimization, restraining *et cetera* is stored in flags, while `MOD` statements are saved as a vector of instances of class `modtask`.
3. The list of required modifications is processed one after another and each one is saved as a new instance of modification class `modi`, which contains all data structures and methods required for application of **one** set of changes. In total, the original group of `atom` instances building the affected residue is replaced by the altered one.
   (a) The residue is checked for validity[6] and whether atoms are available at alternative positions or not. If the latter is true, the second set of coordinates is deleted automatically.
   (b) The required sub-block is loaded from the modifications database file$^{6.1.2}$. If an error occurs at this time, e.g. if the modification cannot be found, the workflow is stopped immediately at this point and proceeds to the next modification[7]. The commands specified in the sub-block for a given combination of force field version, initial residue and modification are transcribed and stored in internal buffer structures. The relative coordinate system is calculated:
      i. Three instances of class `vect` are initialized.
      ii. The vector pointing in `Z` direction has its origin at the anchor point (`ANC` statement). Its direction is defined by two atoms of the original residue (`DVE` statement) and pointing outwards to reduce initial clashes[8]. The vector is set to a length of 1 (normalized).
      iii. Two vectors orthogonal to the `Z` axis and one another are calculated and normalized (`X` and `Y` axes).
   (c) The name of the residue is changed to whatever is stated following directive `NTP`.
   (d) The entries representing the original residue are manipulated:
      i. The `DEL` statements are applied: denoted atoms are deleted from the residue.
      ii. The `REP` statements are applied: the name of the atom given as the first parameter is replaced by the second one. Note, that coordinates are not affected.

[6] It is checked whether the required atoms are given.
[7] This will be mentioned in the logfile.
[8] Most often it is simply the direction of the last remaining bond.

iii. Finally, the ADD statements are processed: the values of XCOORR, YCOORR and ZCOORR are multiplied with the corresponding relative coordinate vectors (see equation 4.1) and this product is added to the anchor point coordinates in order to calculate the absolute coordinates of the new atoms.

Calculation of (absolute) X coordinate

$$\mathrm{x_{newatom}} = \frac{\mathrm{vector_x}}{|\mathrm{vector_x}|} * \mathrm{XCOORR} + \mathrm{x_{anchorpoint}} \tag{4.1}$$

4. File mod_XYZ.pdb is written. In case this option was specified, the original PDB header is retained. The header statements are neither analyzed nor updated and thus the consistency of the final PDB file should be checked manually afterwards.

5. If minimization has not been selected, the backend jumps directly to the generation of the zipped archive. Otherwise, a minimization is performed on the server using GROMACS (version 3.3.3). Sole minimization requires one instance of class minimization, while debumping needs two (pre- and post-run). minimization includes a vector of instances of class ministep which is filled first while parsing the minimization command file$^{6.1.3}$. Afterwards, this vector is worked off from the beginning. ministep buffers information about the product file[9], the command and additional flags required for special cases (restraints and terminal modifications require both additional options as well as an altered minimization protocol - for details see sections 4.1.2.4 and restraints 4.1.2.3). The workflow stalls further step execution if the maximum time limit for a job is exceeded and proceeds as if minimization has not been selected[10]. Job completeness or failure are indicated in the logfile as Job done and Job failed respectively.

6. For each job a zipped archive is generated including all files available for download. This step marks the end of the workflow.

[9] This file is used to check for step completeness.

[10] This means, that the required atoms are added using their relative pre-minimized coordinates, but the overall structure is not minimized.

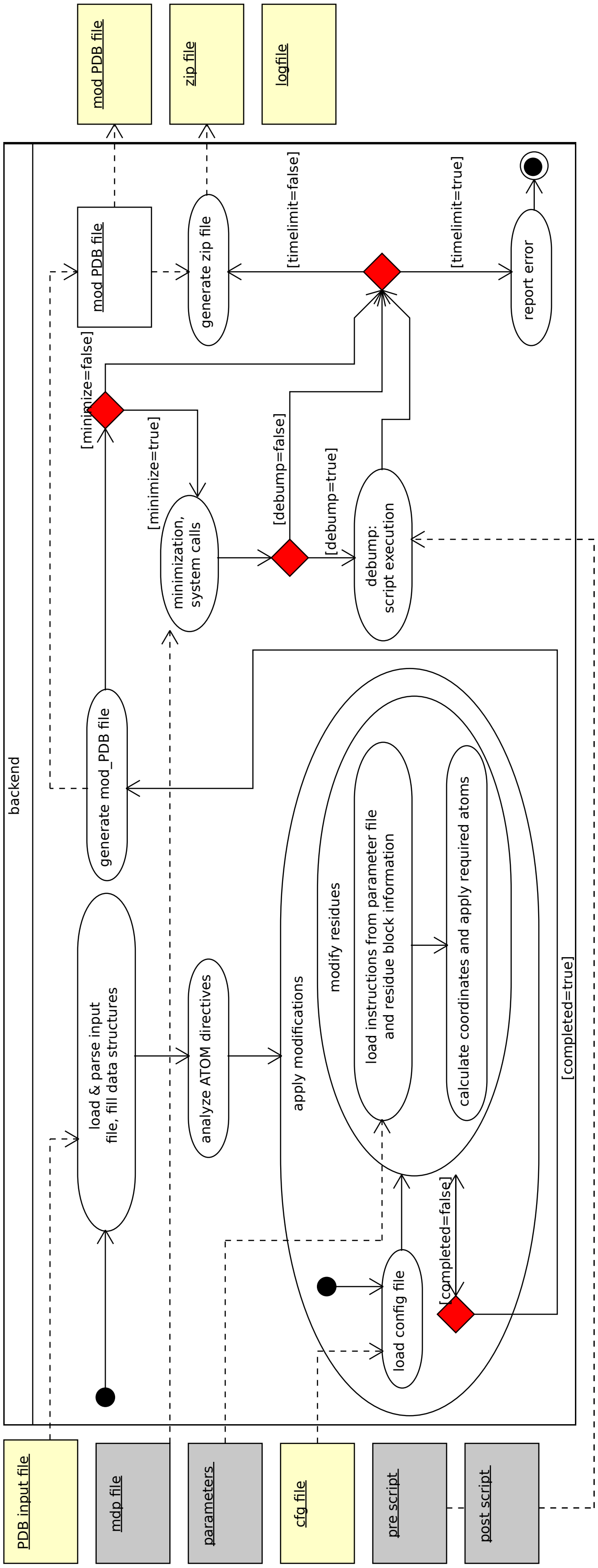


Figure 4.8: Activity diagram (UML) for the backend.

4.1.2.3 Restraints

In order to apply position restraints of a given strength (and restricted to certain atoms only), it is necessary to alter the file `posres.itp`, which is generated by `pdb2gmx`. This file is basically a list of all atom numbers[11] and corresponding restraining strengths for x, y and z axes. Note that there is no possibility provided by the workflow to apply different force constants to the axes. To alter this file, an instance of class `GROMACS_positions_restraints_file` loads and parses `posres.itp`. Atoms added in the previous steps of the workflow are not restrained and therefore deleted. If only the backbone has to be restrained, all side-chain atoms are deleted too. Afterwards, the force constants are replaced by the value specified in the `RESTRAINT` statement in the instruction file.

In the files holding program call information (`par`-files), the flag `<RESTRAINTSSTEP>true` indicates that the restraints file has to be altered prior to this step.

4.1.2.4 Terminal modifications

In GROMACS, terminal modifications are not defined as independent building blocks for each relevant amino acid, but are defined as a list of residue alterations in dedicated files. Thus, the very same steps can be applied to different amino acids since all of them (except proline, for which a unique set of alterations is available) share the same backbone. These instructions are executed by `pdb2gmx`. Since all information about a given molecule is encoded in the topology file, it is, in contrast to other modifications, not necessary to provide these instructions afterwards. On the other hand, it is not possible to use the minimized PDB file from the server as the initial input file since the terminal residue shares its name with unmodified ones.

For terminal modifications, two problems arise due to the implementation of GROMACS and the lack of required setting possibilities:

- The first problem concerns the fact, that GROMACS only supports single atom additions in terms of partial charge groups. Therefore, in order to add bigger groups which shift the charge group enumeration because they include new charge groups, the topology file has to be manipulated after the `pdb2gmx` step. To mark a step to be possibly affected by terminal modifications, the flags `<NTERMINUSMOD>` and `<CTERMINUSMOD>` are set to be `true`. Implementation of the renumbering of partial charge groups depends on whether it is a N- or C-terminal modification:

  - For *N-terminal modifications*, the absolute charge group is stated in the instruction file[6.1.1] and all other charge groups are changed accordingly. It is crucial to keep the same order of atoms as in the corresponding `gro` file, since atoms are only referenced by their number afterwards.
  - In contrast, *C-terminal modifications* require relative shifts refering to the last preceeding charge group, since the absolute value depends on the type of the amino acid. Exactly as explained before, the order of atoms generated by `pdb2gmx` should not be changed.

  For some modifications, it is sufficient to add them in the same charge group as atom `N`, which can be achieved by an (undocumented) `0` in the N-terminal modifications database file.

- The second issue is the implementation of terminal selections in GROMACS. Unfortunately, it has been implemented only to support direct input by the user, making an automatic workflow more difficult. In order to overcome this, an `expect` call has to be made (see also 9.2.1).

[11] Internal numbers of GROMACS, not the ones of the initial PDB file.

4.2 Molecular dynamics

4.2.1 Introduction

Computational calculations have been used for over fifty years [3] to describe molecules at an atomic level and follow their dynamics. These *molecular models* [23, 78, 79] provide additional insights into the character of different compounds in order to amend experimental results or to elucidate aspects which are not even accessible otherwise. In terms of proteins, the focus mainly lies on folding, molecular interactions, partitioning and membrane formation [79]. However, these processes are the net result of a large number of atomic interactions, both bonded and non-bonded. It is a complex issue to find parameters and equations to model these interactions properly, since a more detailed treatment leads in general to higher computational demands. Therefore, the key step in choosing the right formalism for a given problem is to determine the minimum level of accuracy required.

A class of widely used methods for *in silico* simulation of proteins are classical (mechanical) force fields. Although more accurate descriptions exist, these methods are still of high importance because they grant access to timescales long enough to monitor typical biochemical processes. Moreover, they are sufficiently powerful to treat compounds consisting of thousands of atoms[12]. In this work only this type of molecular modeling (MM) is considered. In terms of software, the GROMACS package is one of the most widely used frameworks in computational biophysics, combining a huge set of potential implementations with the ability to use several different force field parameter databases. Currently, all PTMs listed in subsection 9.2.2 are available for GROMOS force fields ffG45a3 and ffG54a7 (see below).

MM is based on providing energy potentials (force fields) describing a given structure (e.g. solved by X-ray crystallography) and is particulary useful for energy minimization (see below) or phase space sampling. However, the major use of force field based methods is in providing a realistic picture of dynamics. Such *molecular dynamics* (MD) means basically to repeatedly alternate between solving Newton's equations of motion and calculating forces acting on an atom. Obviously, treatment of atoms as the smallest (static) entities in such methods is an approximation but the lack of explicit electron clouds, for example, is only important in special cases [78]. The maximum length of a timestep for the integration of Newton's equations of motion directly depends on the highest frequency of all degrees of freedom included in a model. Therefore, it is sometimes straightforward to exclude these if they are not explicitly required for description.

A wide variety of different force fields have been introduced, some of which are designed to fulfill a precisely defined purpose while others are parameterized to serve as a general basis for molecular dynamics simulations. However, independent of their dedication, all classical force fields have been parameterized to match different experimental observables. In order to do so, typically properties of small compounds are either calculated using quantum-mechanics or directly measured experimentally and these values are compared to the ones produced by the force field. Its parameters are then adjusted in an iterative way to match the experiment the best.

It is important to note that besides the simplification of parameters and potentials, another source of potential inaccuracy is parameterization of force fields using small compounds while they are usually applied to large macromolecules such as proteins. However, this is also on the other hand one of the main advantages of these methods as it renders them general and transferable.

[12] Compared to *quantum mechanics*, which can be used to describe small systems in detail.

4.2.2 Potentials and parameters

As mentioned above, it is necessary to provide parameter sets for different types of atoms and the interactions between them in order to model molecules at atomic resolution. Often, there are many different terms available for a particular interaction. However, since sophisticated equations increase computational demand, a compromise is often required. For the following explanations, the implementation in GROMACS (as defined in [78]) is considered.

4.2.2.1 Coulomb interactions

Interaction between two charged particles is described by the **Coulomb potential**

$$V_c(r_{ij}) = \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{\varepsilon_r r_{ij}} \tag{4.2}$$

where $q_{i,j}$ is the charge of atoms i and j, r_{ij} is the distance between them and ε is the dielectric constant. Since the strength of this interaction drops off linearly with increasing distance, this term has to be considered also for partners which are relatively far apart.

For practical reasons, it is often further approximated for longer distances above a certain cut-off e.g. by usage of a reaction field term, which simply postulates a uniform surrounding of a given value of dielectric constant[13].

Coulomb potential (reaction field)

$$V_{crf} = \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{\varepsilon_r r_{ij}}\left[1 + \frac{\varepsilon_{rf}-\varepsilon_r}{2\varepsilon_{rf}+\varepsilon_r}\frac{r_{ij}^3}{r_c^3}\right] - \frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{\varepsilon_r r_c}\frac{3\varepsilon_{rf}}{2\varepsilon_{rf}+\varepsilon_r} \tag{4.3}$$

All pair interactions, for which r_{ij} is bigger than the cut-off radius r_c are calculated by an extra term including ε_{rf} which is the dielectric constant for the reaction field.

4.2.2.2 Lennard-Jones interactions

The **Lennard-Jones potential**

$$V_{LJ}(r_{ij}) = \frac{C_{ij}^{12}}{r_{ij}^{12}} - \frac{C_{ij}^{6}}{r_{ij}^{6}} \tag{4.4}$$

covers interactions based on induced dipoles and short-range repulsion (van-der-Waals interactions). It contains an attractive (C_{ij}^6/r_{ij}^6) and a steep repulsive (C_{ij}^{12}/r_{ij}^{12}) term, due to overlap of the electron clouds. The strength of dipoles and ultimately of the attractive part of the interaction depends on the number of electrons and their configuration. Thus, it differs for each defined pair of N elements leading to a NxN-matrix of required interaction parameters. However, they can be approximated by calculating their **geometric averages**

$$\begin{aligned} C_{ij}^{(6)} &= (C_{ii}^{(6)}C_{jj}^{(6)})^{\frac{1}{2}} \\ C_{ij}^{(12)} &= (C_{ii}^{(12)}C_{jj}^{(12)})^{\frac{1}{2}} \end{aligned} \tag{4.5}$$

which strongly reduces the number of initial parameters.

[13] Other, more sophisticated, methods to include long-range electrostatic interactions are the Ewald summation and **p**article **m**esh **E**wald (PME).

Another possibility is to express the LJ-potential as

$$V_{LJ}(r_{ij}) = 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right) \tag{4.6}$$

and to use the **Lorentz-Berthelot rules**

$$\begin{aligned}\sigma_{ij} &= \frac{1}{2}(\sigma_{ii} + \sigma_{jj}) \\ \epsilon_{ij} &= (\epsilon_{ii}\epsilon_{jj})^{\frac{1}{2}}\end{aligned} \tag{4.7}$$

Note, that there are more sophisticated (but also computationally more demanding) methods available to compute both van-der-Waals and Coulomb interactions. Non-bonded interactions require structural information since it is not defined by the primary structure, which atoms are within a certain proximity of another atom and thus considered to have an effect. One possible solution is to keep track of distances between atoms in so-called neighbour lists.

4.2.2.3 Bonded interactions

Covalent bonds between atoms are very stable in comparison to the nb-interactions described above. They result from sharing one or more electrons between adjacent atoms and solely determine the primary structure of a protein. In order to specify them, the following four potentials are used.

Bonds

$$V_b(r_{ij}) = \frac{1}{2}k^b_{ij}(r_{ij} - b_{ij})^2 \tag{4.8}$$

This harmonic potential defines the change in energy resulting from *bond stretching*. The discrepancy of the calculated bond length r_{ij} between atoms i and j and the minimal energy bond length for this pair of atoms, b_{ij}, gives rise to increased energy, proportionally to the Hookean spring constant k_{ij}. Therefore, a database with all supported covalent bond types, including the corresponding spring constants and equilibrium bond lengths, have to be stated (for example, see subsection 9.4.2 for the list of bonded parameters in ffG45a3 GROMOS force field).

Angles

$$V_a(\theta_{ijk}) = \frac{1}{2}k^\theta_{ijk}(\theta_{ijk} - \theta^0_{ijk})^2 \tag{4.9}$$

The same remarks as for the bond-stretching harmonic potential can be made for the *angle potential*, with the exception, that here three linearly linked atoms have to be considered. The properties of an angle built by atoms i, j and k depends on the atom types involved. Again, a discrepancy between the minimum angle and the observed one leads to a higher energy. A list of all available harmonic angle parameters for ffG45a3 can be found in subsection 9.4.2.

Improper dihedrals

$$V_{id}(\xi_{ijkl}) = \frac{1}{2}k_\xi(\xi_{ijkl} - \xi_0)^2 \tag{4.10}$$

These artificial dihedral angles are introduced to ensure either planarity (e.g. in a benzene ring) or to prevent chiral molecules (e.g. amino acids) from changing their conformation. The nature of this potential is described by a harmonic oscillator. In ffG45a3, only three types have been defined[9.4.2]: `gi_1` for planar groups, `gi_2` for tetrahedral centres and `gi_3` for the heme iron atom.

Dihedrals

$$V_d(\phi_{ijkl}) = k_\phi(1 + \cos(n\phi - \phi_s)) \tag{4.11}$$

When four atoms are connected linearly, rotation around the bond in the middle has to be described. If the atoms i, j, k and l build a chain, the dihedral angle is the one between the two planes defined by ijk and jkl. The equation above is the one used to implement a periodic type of dihedrals. k_ϕ is the force constant, n the periodicity[14] and ϕ_s is the angle at which the minimum is reached. A list of dihedral parameters for ffG45a3 is given in subsection 9.4.2.

4.2.2.4 Position restraints

In some cases, such as partial minimization, it is useful to force all atoms or a subset to stay near their initial positions. Therefore, an artificial potential can be applied, whose parameters can be chosen as required.

Position restraints potential

$$V_{pr}(r_i) = \frac{1}{2}k_{pr}\,|r_i - R_i|^2 \tag{4.12}$$

The restraints potential is again modeled as a harmonic one, with one force constant for each axis of the coordinate system. Note, that this functionality is available for the present workflow in order to minimize the energy of the modified residues while keeping the rest of the structure mostly immobile. The value of the force constant chosen by the user is currently used for all three directions.

4.2.3 Parameterization of PTMs

In order to find new parameters for the set of supported PTMs, we followed the general GROMOS philosophy: Big groups are decomposed into smaller fragments or even just chemical functions in order to reduce the number of required atom types, bonds, angles *et cetera.* The number of required parameters can be lowered significantly due to this rule. However, it should also be pointed out that by these sub-summation, additional errors are introduced since the influence of the proximal atoms is neglected.

Except for some special cases (for details, see description in section 5), the assignment of bonded parameters and the selection of atom types is straightforward. Therefore, the central step in parameterizing new residues is the specification of appropriate partial charges, since they most dominantly reflect the chemical nature of the compound and are important for its interaction behaviour.

4.2.3.1 Atom names

Atom naming conventions:

- The backbone atoms are a special case: Those which are part of the peptide bonds are simply named after the element type of the atom: `N`, `H`, `C` and `O` (see also subsection 3.1).
- Other heavy atoms are named as follows:
  1. The first letter is always the element type.
  2. The second one is taken from the Greek alphabet according to its position as seen from the alpha carbon: A (alpha), B (beta), G (gamma), D (delta), E (epsilon), Z (zeta), H (eta), T (theta), I (iota) *et cetera.*

[14] Meaning, how often the potential reaches its minimum during a complete rotation.

3. If necessary, the third position can be a number in order to distinguish two atoms. This is necessary, since atom names are used as unique identifiers.

- Hydrogen atoms are named after the heavy atom they are attached to. If necessary, a fourth letter can be added in order to specify two hydrogens properly. For example, if nitrogen `NH1` is bound to two hydrogens, they would be named `NH11` and `NH12` respectively.

Note that atoms added to the terminal ends of proteins are not named by this scheme. Moreover, in some cases this procedure could not be applied because `pdb2gmx` automatically renames certain atoms which would appear when following this specification. Therefore, the next letter has been used in these cases.

4.2.3.2 Exclusions

In general, exclusions are made for interactions of an atom i and atoms covalently linked to i via one, two or three covalent bonds, because those are modeled already by the covalent interactions (see above). This means that the LJ-interaction e.g. between atoms i and i+2 are not included when determining forces.

Moreover, it is sometimes required to add additional exclusions (see the `[ exclusions ]` block in subsection 6.3.4), in order to avoid non-bonded contributions affecting correct modeling of a group or molecule. If a system shows unexpected behaviour, exclusions might be an easy way to overcome this.

4.2.3.3 Dihedrals

Dihedrals are defined by four atom names (see above) but in most cases, they are specified only by the types of the two atoms in the middle. A dihedral entry (see 9.4.2) contains all the information required to calculate the potential listed above. Moreover, it is possible to use more than one dihedral for a particular bond (see phospho-groups). Usually, heavy atoms are preferred over hydrogens to define proper dihedrals (see `[QME]`).

4.2.3.4 Impropers

For impropers defining stereochemistry in GROMACS, four entries (ijkl) are required: i stays always the same (atom in the middle) and there are six possibilities to order the remaining three. However, there are only two different cases since only the order of k and l matters: jkl - klj - ljk and jlk - lkj - kjl, each representing one stereochemical conformation.

4.2.3.5 Partial charges

It is a common approach to set the charges of those atoms first, which are accessible from the surrounding environment (in regard of their interaction potential *et cetera*) and to adapt the value of the inner atoms afterwards to get the required overall charge. All charge groups are set to ensure, that the sum of their partial charges is an integer number. Most often it is `0`, but also `+1` e.g. in case of lysine and `-1` or `-2` e.g. for phospho-groups.

One important aspect while parameterizing partial charges, is to come up with meaningful charge groups. For some groups, when electrons are shared to a significant amount by more than just a few atoms, charge groups cannot be decomposed (e.g. arginine side chain). In these cases, one can either search the literature for parameters or use similar groups as the starting point.

4.2.4 GROMOS ffG45a3 and ffG54a7

Force field G45a3 has been published in 2001 in order to overcome the limitations of the previous force field G45a1 when it comes to the proper description of lipids [68]. To achieve this, lipids have been added to the set of small molecules whose properties the force fields were fit against. It has been demonstrated that G45a3, while reproducing densities, heats of vaporization and free energies of hydration for alkanes quite well, also describes well protein secondary and tertiary structure and dynamics [74].

Parameterization of the G53a6 force field in 2004 was focused on the proper treatment of solvation free energies for amino acids, because these are of high importance in protein folding. The strategy was the following: after modeling small molecules in cyclohexane in order to match thermodynamic properties (which gave rise to force field version G53a5), the partial charges of the resulting parameters were adjusted to fit to observed solvation free energies in water [56]. Comparison with G45a3 revealed a similar tendency to retain the overall structure of proteins [55].

Force field G54a7 is based on G53a6 and improves four aspects, of which the most important one is the change of certain van-der-Waals parameters and torsional-angle energies for peptide backbone dihedrals in order to improve agreement of simulated protein secondary structure and the one determined experimentally [67]. The main effect are stronger hydrogen-bonds in the backbone, granting stable structures while still matching experimental NOE intensities and ^{3}J-coupling values.

4.2.5 Minimization

Energy minimization can be used, to reduce the overall potential energy of a molecule as well as, in terms of this workflow, to "relax" the attached groups in the context of the whole compound. Minimization on the server is performed *in vaccuo* by default. Moreover, in order to retain the initial coordinates of the atoms, restraints can be used to ensure they will not change their positions too drastically.

Listing 4.4: Excerpt of the servers `mdp` file

```
integrator       = steep
emtol            = 1.0

nstlist          = 3
ns_type          = simple
rlist            = 1.4
coulombtype      = cut-off
rcoulomb         = 1.4
rvdw             = 1.4
constraints      = none
pbc              = no
```

4.2.6 Testing of parameters

After re-checking the parameter sets carefully, short simulations were run to exclude typing errors, missing bonds *et cetera* for all building blocks in both force fields supported. In the future, *thermodynamic integration* (TI) [11, 56] will be used to compare the hydration free energy of the modified residues with experimental values. This approach has also been used by Oostenbrink *et al.* [56] to validate the parameters of ffG53a6 (see above). However, experimental data regarding hydration free energy is limited and not available for all the compounds parameterized. Therefore, molecules for which those experiments have been done, will be parameterized using the very same methods as in the case of the non-canonical amino acids. If initial parameters do not reliably reproduce the experimentally observed behaviour, they have to be adapted.

4.2.6.1 Thermodynamic integration

It is not a trivial task to calculate the free energy of a system: in order to treat the entropic contribution properly, in principle the whole phase space for a given system [11] has to be covered. Since this is computationally too demanding, other methods have been developed to approximate free energy (differences) which are based on *coupling parameter approaches.* For the present work and the on-going validation of the parameter sets, *termodynamic integration* (TI)is used to calculate hydration free energy.

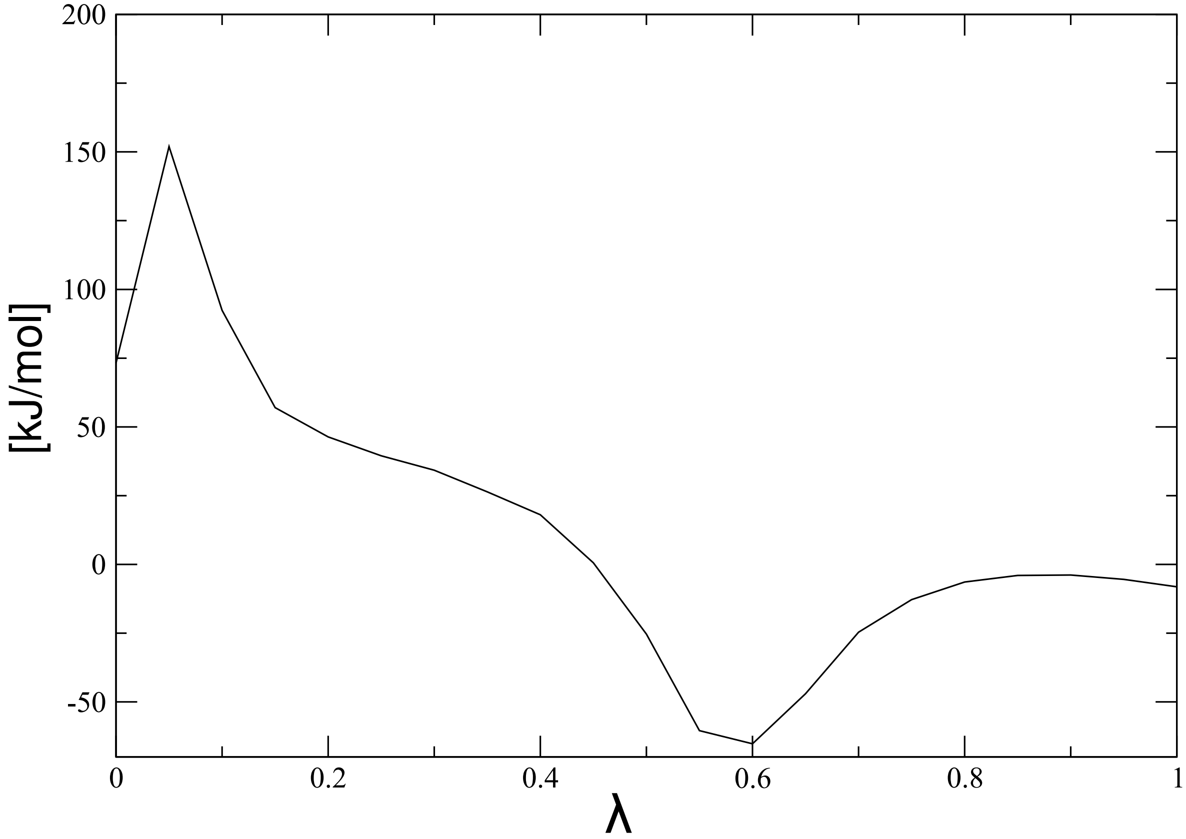


Figure 4.9: Example of a TI graph. On the y-axis, the average derivative of the Hamiltonian $\langle \frac{\partial H}{\partial \lambda} \rangle$ in kJ/mol at a given λ point ensemble, while the x-axis, ranging from 0 to 1, covers all λ steps. The resulting curve is integrated to get the difference in free energy of solvation between states 0 and 1.

Change in free enthalpy

$$\Delta G = \int_0^1 \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_\lambda d\lambda \tag{4.13}$$

In typical TI calculation, the state 0 represents non-bonded interactions fully turned off, while in state 1 they are present to a normal extent. In order to calculate ΔG, a series of simulations has to be set up to calculate the mean derivative of the Hamiltonian with respect to λ at several λ points. In general, the more sampling points are available, the more accurate the calculation is.

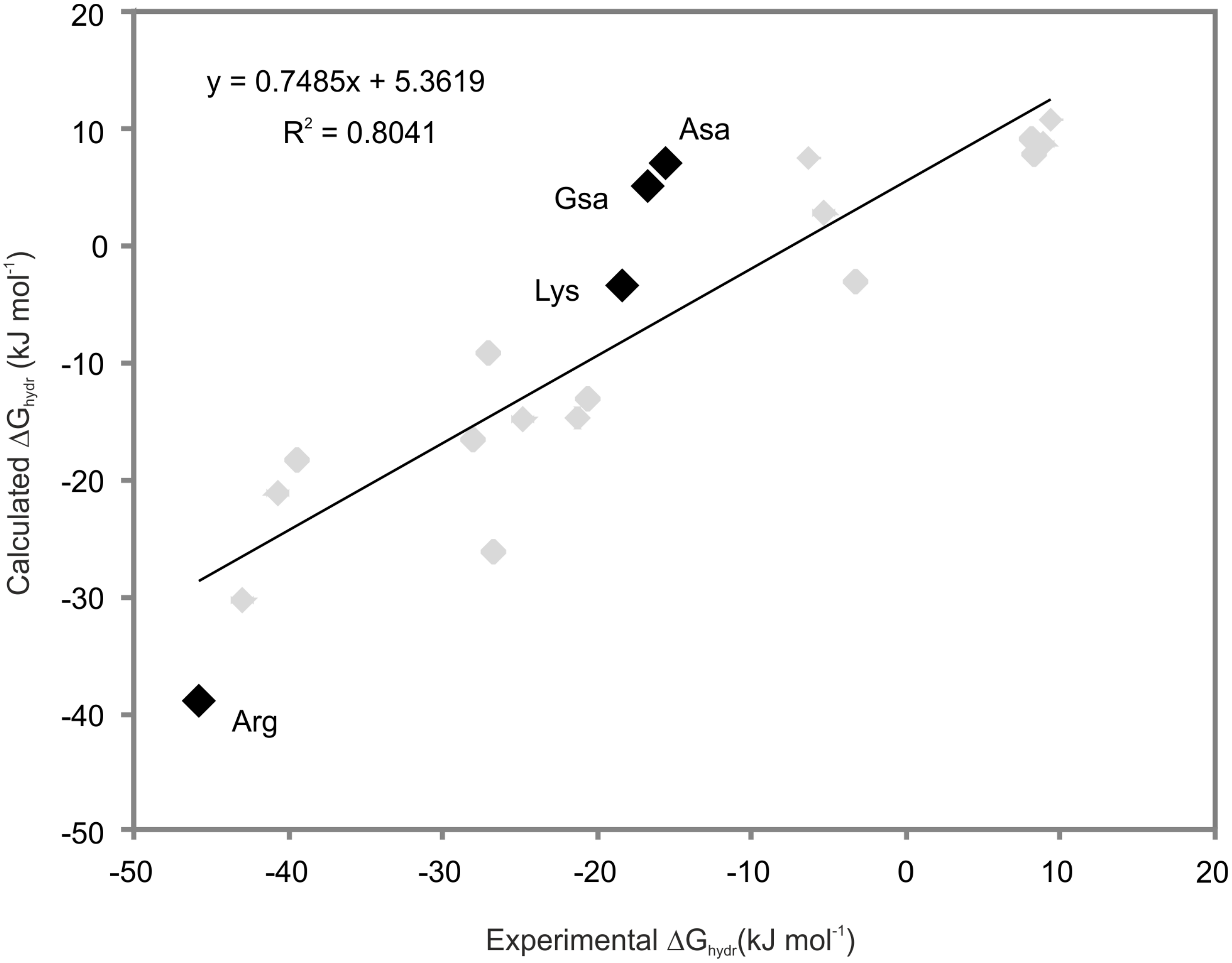


Figure 4.10: Comparision of calculated and experimental ΔG_{solv} of amino acids and carbonylated derivatives, where the line indicates perfect match. Modified from [61]. In a typical parameterization procedure, partial charges of the building blocks can be fine-tuned in order to match the experimental values best. This procedure will in the future be applied for the extended amino acid alphabet where possible.

4.3 Technical specifications

Hardware		
Function	**Specification**	**Details**
Processor	2x QuadCore Intel(R) Xeon(R)	2.67 GHz
Disk	-	452 GB
RAM	-	24 GB
Server		
Name	coil	dedicated server
Operating system	Red Hat Linux (x64)	Version 2.6 (or higher)
Server-side scripts	PHP: Hypertext Preprocessor	Version 5.1 (or higher)
Database	MySQL	Version 12.1 (or higher)
Backend	C++ binary	Version 1.1 (or higher)
Client		
Graphs	jpgraph	Version 3.5.0b1
JS library	jQuery	Version 1.2.6
JS menu	superfish	Version 1.4.8
JS menu	scriptaculous	Version 1.9.0
Molecule representation	Jmol	Version 12 (or higher)

Table 4.2: Technical specifications of Vienna-PTM (July 2012).

CHAPTER 5

PTMs in GROMOS 45a3 force field

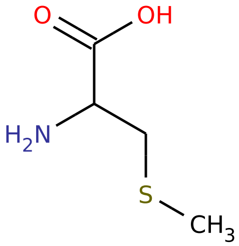
This chapter gives particularly important parameterization details for all PTMs developed for the GROMOS 45a3 force field [60]. All pictures were made by use of Marvin Sketch [14]. For extended information regarding a particular compound, see hyperlinks to PubChem [13] and ChemSpider [66] databases. Names of atoms$^{4.2.3.1}$, chemical functions and force field parameters are highlighted (e.g. ga_9). For details regarding the implementation of new residues, see section 4.2 and section 6.3 for a description of the parameter file formats respectively. Note, that no new parameters have been introduced for modeling the post-translationally modified amino acids but rather pre-existing parameters from a given force field were used. The modifications are named as such, that the first letter represents the original residue, while the other two are used to describe the type of modification. However, there are exceptions - e.g. if a particular PTM has been defined previously or a name is already in use.

5.1 Methylations

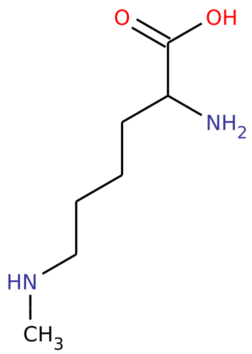
5.1.1 Single methylations

Name: S-methylcysteine
Abbreviation: [CYM]
ChemSpider: 196235 PubChem: 24417

Notes: Partial charges for -S-CH3 were chosen according to the terminal group of Methionine, since the local chemical structure is the same. When possible, this strategy has been preferred to assign the partial charges in order to be consistent with the canonical amino acid. The total charge of the side-chain is zero.

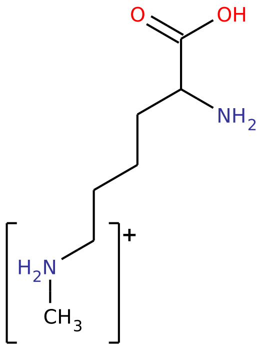


Name: N^6-methyllysine
Abbreviation: [KMN]
ChemSpider: 144469 PubChem: 164795



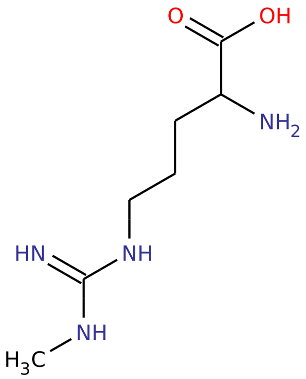
Notes: For tetrahedral nitrogens[1], the angles do not sum up to 360°. Therefore, angles ga_9, ga_10, ga_12 and ga_14 have been used for those. CH$_2$ groups are more open and therefore ga_14 was used (all others are of type ga_12). Partial charge difference has been increased due to the free electron pair at atom NZ which has sp^2 character here [54].

Name: N^6-methyllysine (charged)
Abbreviation: [KMC]
ChemSpider: 144469 PubChem: 164795



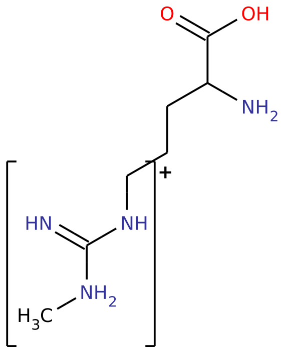
Notes: Same remarks for the N as for [KMN], but in contrast the sum of the individual partial charges of the terminal group is +1 and the atom type is NL instead of NT or N (both in uncharged versions). The key change is that of the N: from -0.88 to +0.104.

Name: ω-N-methylarginine
Abbreviation: [RMN]
ChemSpider: 117259 PubChem: 4366



Notes: The type of the N, to which the additional CH$_3$-group is attached, was changed to NE (like the preceding N in arginine) and its partial charge was shifted to match the chemical nature best. The overall charges of charge groups 3, 4 and 5 are zero.

Name: ω-N-methylarginine (charged)
Abbreviation: [RMC]
ChemSpider: 117259 PubChem: 4366

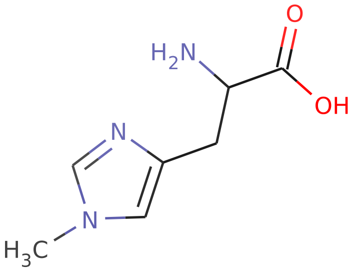


Notes: There is only one big charge group (number 2), consisting the whole end of the residue (since the +1 charge is distributed). Moreover, because of the additional H, in constrast to [RMN], atom NH2 remains type NZ, the angles around that atom sum up to 360° and improper NH2 HH2 CT CZ gi_1 ensures planarity.

[1] In contrast to planar ones as in [QME].

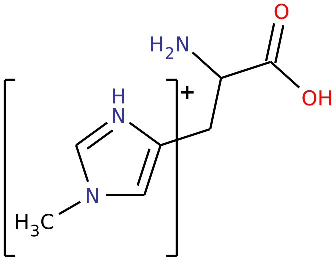
Name: 1'-methylhistidine
Abbreviation: `[H1M]`
ChemSpider: 312567 PubChem: 92105

Notes: One big charge group has been used, because individual chemical groups do not sum up to zero. Exclusions have been introduced4.2.3.2 for atom `CZ` with both `CG` and `ND1`.



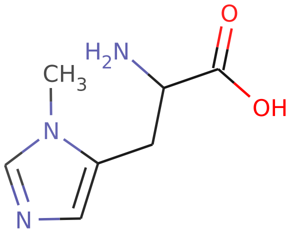
Name: 1'-methylhistidine (charged)
Abbreviation: `[H1C]`
ChemSpider: 312567 PubChem: 92105

Notes: Total charge of residue terminal group is +1. Additional improper needed for the extra H compared to `[H1M]`.



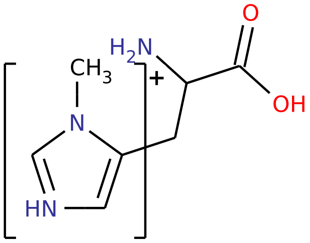
Name: 3'-methylhistidine
Abbreviation: `[H3M]`
ChemSpider: 491986 PubChem: 64969

Notes: Because of spatial proximity, additional exclusions are required for the methyl group `CE2` (with `CB`, `CD2` and `NE3` respectively) in order to avoid non-bonded interactions for those pairs. An improper dihedral is used to keep `CE2` in one plane with the ring.



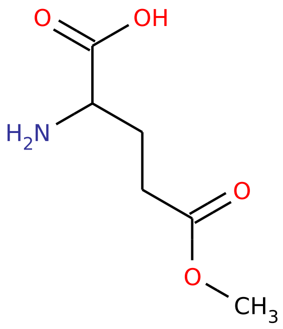
Name: 3'-methylhistidine (charged)
Abbreviation: `[H3C]`
ChemSpider: 491986 PubChem: 64969

Notes: In contrast to `[H3M]`, one H has been added and partial charges have been changed in order to give a +1 overall charge. Major change: `-0.58` to `0.3` at `NE2`, which is the anchor point for the H.



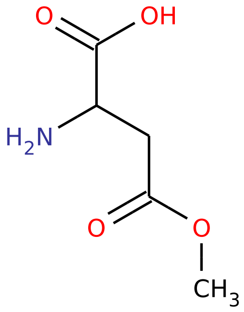
Name: methylglutamate
Abbreviation: `[EME]`
ChemSpider: 19970565 PubChem: 68662

Notes: The type of atom `OE2` (anchor point for methyl group `CZ`) was changed to `OA` and that of `OE1` to `O`. This was necessary, since a distinct type is used for both `O` atoms in the terminal carboxy group (`OM`), which is no longer present. Partial charges were assigned according to compound `[DPPC]`.



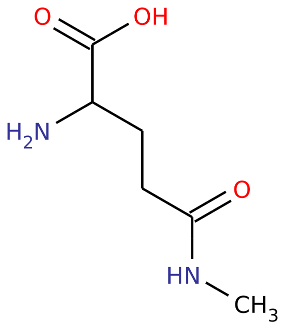
Name: methylaspartate
Abbreviation: `[DME]`
ChemSpider: 92764 PubChem: 102699

Notes: Same remarks as for `[EME]`.



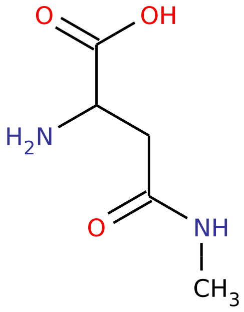
Name: methylglutamine
Abbreviation: `[QME]`
ChemSpider: 388957 PubChem: 25203300

Notes: Parameters have been set according to peptide-bond, which is the same from a chemical point of view. Consequentially, the type of the nitrogen has been changed from `NT` to `N`. For consistency, partial charges were also chosen and grouped as in peptide-bonds.



Name: methylasparagine
Abbreviation: `[NME]`
ChemSpider: 90708 PubChem: 100393

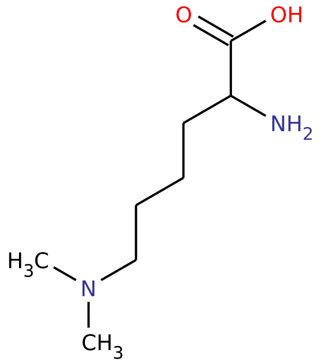
Notes: Same remarks as for `[QME]`.



5.1.2 Double methylations

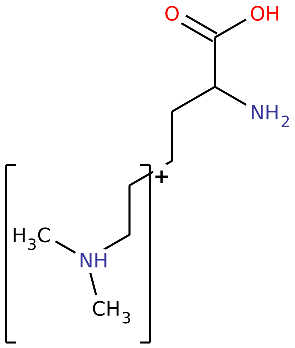
Name: N^6-N^6-dimethyllysine
Abbreviation: [K2M]
ChemSpider: 167778 PubChem: 193344

Notes: Same remarks for the N as for [KMN]. The overall charge of the modified group stays the same, but distribution was changed due to H ↔ CH_3 swap (N less negative). The exact values have been taken from ammonium.



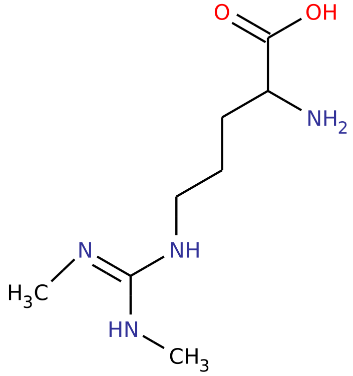
Name: N^6-N^6-dimethyllysine (charged)
Abbreviation: [K2C]
ChemSpider: 167778 PubChem: 193344

Notes: In contrast to [K2M], the type of N is NL (not NT) because of the additional substituent. The total charge of the model group is +1.



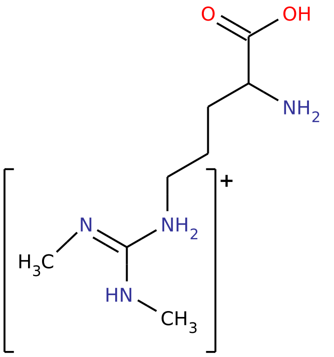
Name: (symmetric) dimethylarginine
Abbreviation: [RSM]
ChemSpider: 147942 PubChem: 169148

Notes: The terminal group was split into 3 charge groups, which sum up to a total charge of 0. The types of NH1 and NH2 have been changed to NE as in [RMN]. Atom NH1 has an additional hydrogen as substituent compared to NH2, since it is the neutral version. Angle ga_26 has been used for consistency reasons.



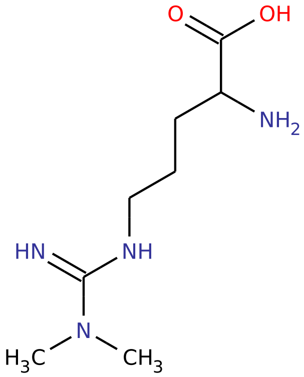
Name: (symmetric) dimethylarginine (charged)
Abbreviation: [RMS]
ChemSpider: 147942 PubChem: 169148

Notes: Same remarks as for [RSM], but the charge groups 3, 4, 5 and 6 have been merged here and +1 charge is distributed. Both terminal nitrogen atoms carry an extra hydrogen.



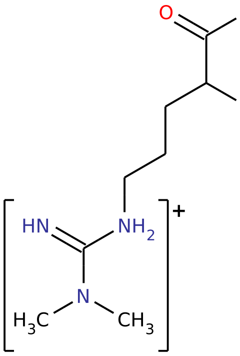
Name: (asymmetric) dimethylarginine
Abbreviation: [RAM]
ChemSpider: 110375 PubChem: 123831

Notes: Atom NH2 remains type NZ, but the individual charges of this group (number 5) are adjusted to match those of DNA. The total charge is 0. All angles around the methylated nitrogen are of type ga_27 to fortify natural conformation.



Name: (asymmetric) dimethylarginine
Abbreviation: [RMA]
ChemSpider: 110375 PubChem: 123831

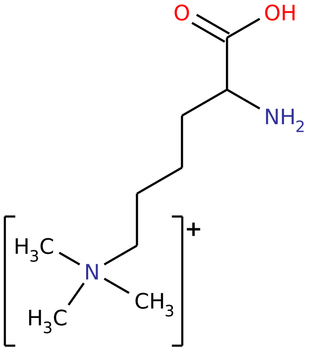
Notes: A total charge of +1 is distributed over group 2. Remarkably, the charge of atom NH2 is very low to avoid particular interactions there. The angles around NH2 are the same as in [RMS].



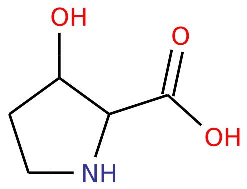
5.1.3 Triple methylations

Name: N^6-N^6-N^6-trimethyllysine
Abbreviation: [K3C]
ChemSpider: 389121 PubChem: 440121

Notes: Same remarks for the N as in [KMN]. No improper required here for NZ CH1 CH2 CH3, since the angles of all four atoms sustain the conformation. The overall charge is +1, evenly distributed over N, the preceding CH_2 group and all three methyl-groups.

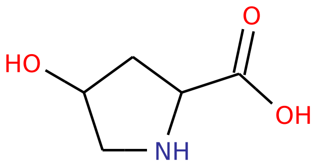


5.2 Hydroxylations



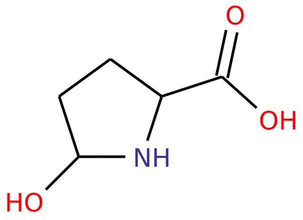
Name: 3'-hydroxyproline
Abbreviation: [P3H]
ChemSpider: 486216 PubChem: 559314

Notes: There is also a second version available (called [PH3], R conformation), which differs just in one improper. Stereochemical versions[4.2.3.4]: CB CG2 CA OG1 gi_2 in [P3H] (S conformation) and CB CG2 OG1 CA gi_2 in [PH3].



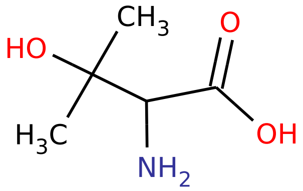
Name: 4'-hydroxyproline
Abbreviation: [HYP]
ChemSpider: 802 PubChem: 825

Notes: This residue already exists in the original RTP file[6.3.4]. As for [P3H], an additional stereoversion (R conformation) has been introduced: [HY2].



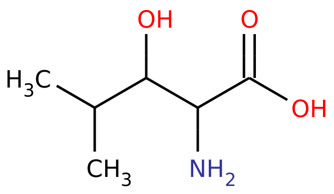
Name: 5'-hydroxyproline
Abbreviation: [P5H]
ChemSpider: 10629201 PubChem: n/a

Notes: Same remarks as for [P3H]. An exclusion was required for HE and N. Its stereoversion is [PH5]. Partial charges for all proline hydroxylations follow standard values. The O and the H group together with the particular anchor C. Stereochemical version (R conformation): [PH5].



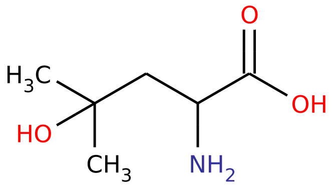
Name: 3'-hydroxyvaline
Abbreviation: [V3H]
ChemSpider: 244525 PubChem: 277794

Notes: The charge of CB was changed to 0.15 while those of CG1 and CG2 stay 0.00. One important change is that of atom CB to type CHO, which is a tetrahedral aliphatic carbon[9.4.1] (there is no improper required here).



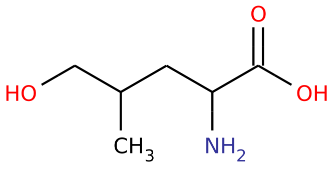
Name: 3'-hydroxyleucine
Abbreviation: [L3H]
ChemSpider: 244507 PubChem: 277776

Notes: Standard charge distribution for a -CHn-OA-H group has been used. Around CB and CG2, angles ga_12 and ga_14 have been used respectively. The additional stereochemical version (S conformation) is [LH3].



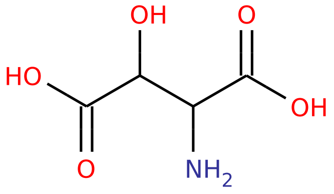
Name: 4'-hydroxyleucine
Abbreviation: [L4H]
ChemSpider: n/a PubChem: n/a

Notes: Same remarks as for [L3H].



Name: 5'-hydroxyleucine
Abbreviation: [L5H]
ChemSpider: n/a PubChem: n/a

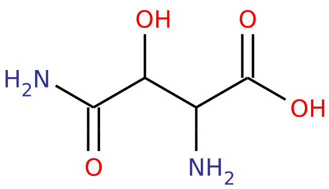
Notes: Same remarks as for [L3H]. Stereochemical version (S conformation): [LH5].



Name: 3'-hydroxyaspartate
Abbreviation: [D3H]
ChemSpider: 5232 PubChem: 14463

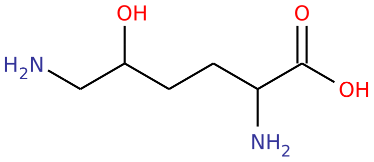
Notes: The hydroxy and the carboxy group were modeled separately and therefore standard charges have been used. Stereoversion (R conformation): [DH3].

Name: 3'-hydroxyasparagine
Abbreviation: [N3H]
ChemSpider: 3670287 PubChem: 152191



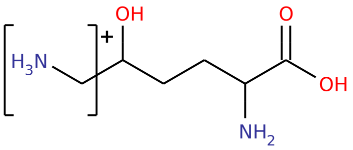
Notes: Standard partial charges for hydroxy, amino and carboxy groups have been used. An additional improper is required at atom CB. Stereoversion (R conformation): [NH3].

Name: 5'-hydroxylysine
Abbreviation: [K6H]
ChemSpider: 1002 PubChem: 1029



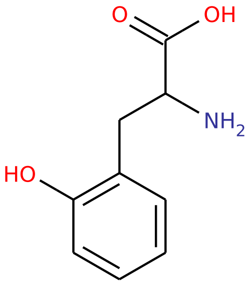
Notes: Same remarks as for [L3H]. A stereoversion (R conformation) exists: [KH6].

Name: 5'-hydroxylysine (charged)
Abbreviation: [KHP]
ChemSpider: 1002 PubChem: 1029



Notes: Same remarks as for [L3H]. The +1 overall charge is distributed over the H atoms (0.248 each), atom NZ (0.129) and CE2 (0.127). The stereoversion (R conformation) of this building block is [KPH].

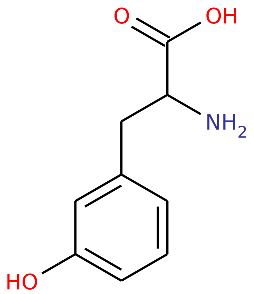
Name: 2'-hydroxyphenylalanine
Abbreviation: [F2H]
ChemSpider: 82607 PubChem: 91482



Notes: Charge distribution as usual. An additional improper CD1 CG CE1 OE3 gi_1 is required to keep OE3 in the same plane as the ring (cf. hydroxy-group in [TYR]). Four additional exclusions have been added due to atom OE3.

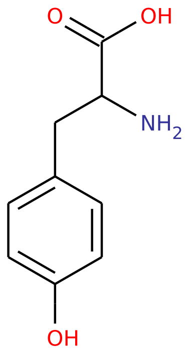
Name: 3'-hydroxyphenylalanine
Abbreviation: `[F3H]`
ChemSpider: 12509 PubChem: 13052

Notes: Same remarks as for `[F2H]`.



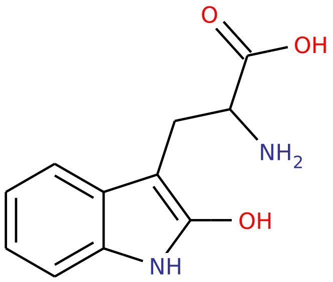
Name: 4'-hydroxyphenylalanine / tyrosine
Abbreviation: `[TYR]`
ChemSpider: 1121 PubChem: 6057

Notes: Same remarks as for `[F2H]`.



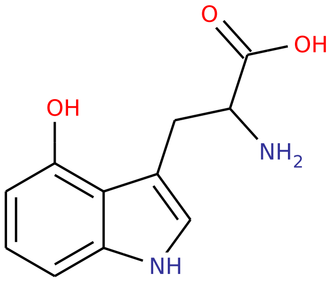
Name: 2'-hydroxytryptophan
Abbreviation: `[W2H]`
ChemSpider: 2627909 PubChem: 3382782

Notes: Standard charge distributions were used for the hydroxy-group. Improper and exclusions as in `[F2H]`.



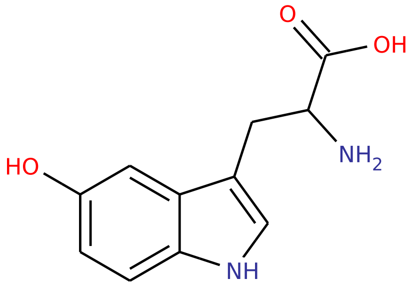
Name: 4'-hydroxytryptophan
Abbreviation: `[W4H]`
ChemSpider: 512674 PubChem: 589768

Notes: Same remarks as for `[W2H]`.



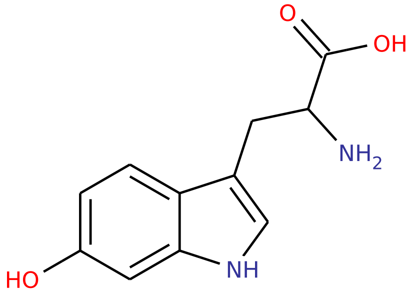
Name: 5'-hydroxytryptophan
Abbreviation: [W5H]
ChemSpider: 141 PubChem: 144

Notes: Same remarks as for [W2H].



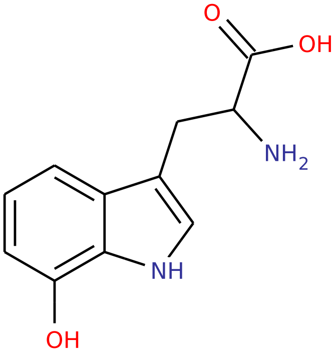
Name: 6'-hydroxytryptophan
Abbreviation: [W6H]
ChemSpider: 8074974 PubChem: n/a

Notes: Same remarks as for [W2H].

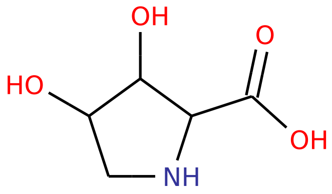


Name: 7'-hydroxytryptophan
Abbreviation: [W7H]
ChemSpider: 2043769 PubChem: n/a

Notes: Same remarks as for [W2H] but additional exclusions are required due to the spatial neighborhood between the hydroxy and the amino group: NE1 HH2 and OH2 HE1.

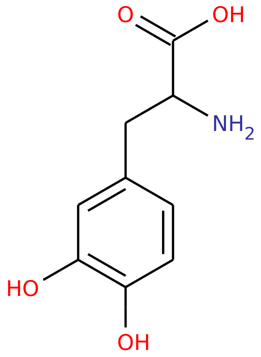


5.2.1 Double hydroxylations



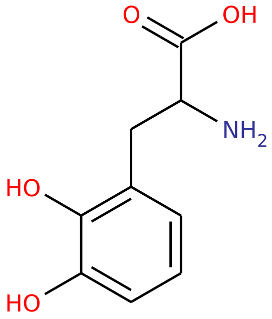
Name: 3',4'-dihydroxyproline
Abbreviation: [PHH]
ChemSpider: 494156 PubChem: 568381

Notes: Both hydroxy-groups, together with either CB or CG2, were designed as in [P3H] with regard to partial charges and charge groups. Moreover, they lead to two additional chiral centers which require impropers of type gi_2.



Name: 3',4'-dihydroxyphenylalanine
Abbreviation: [HTY]
ChemSpider: 813 PubChem: 836

Notes: The initial template is either [PHE] or [TYR] and the result is in both cases [HTY]. The hydroxy groups, together with their attachment atoms in the benzene ring, are treated on their own in terms of partial charges (standard hydroxy groups).



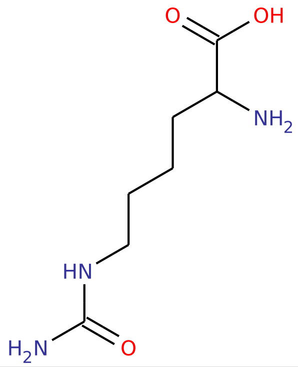
Name: 2',3'-dihydroxyphenylalanine
Abbreviation: [F23]
ChemSpider: 9508015 PubChem: 11333069

Notes: Same remarks as for [HTY].

5.3 Carboxylations / Carbamylations

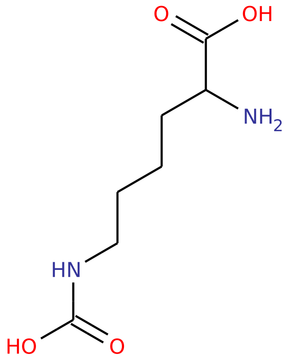
Name: carbamylated lysine
Abbreviation: [KAM]
ChemSpider: n/a PubChem: n/a

Notes: The link between the canonical residue and the attached modification was designed as in the peptide bond, since it is similar from the chemical point of view. The resulting structure was split up in 3 different groups. For the angle OI1 CH NI2, type ga_29 was used to come closer to urea properties (even if description does not match). Atom names are not consistent because of "GROMACS naming problem"[4.2.3.1].



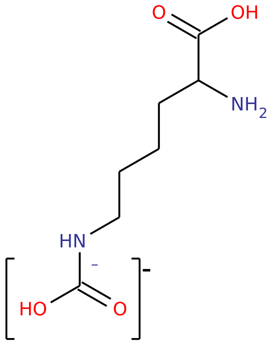
Name: carboxylysine (neutral)
Abbreviation: [KCN]
ChemSpider: n/a PubChem: 17754054

Notes: The overall charge is 0, the terminal carboxy-group is protonated (one O and one OA). The angles around atom CZ are twice ga_30 and once ga_20, which sums up to 360°. These angles had to be defined carefully in order to avoid mutual compression. Atom names are not consistent because of "GROMACS naming problem"[4.2.3.1].



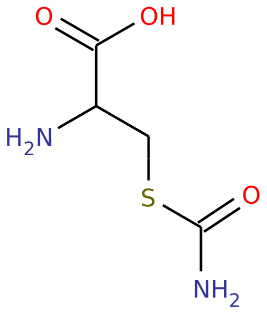
Name: carboxylysine (charged)
Abbreviation: [KCA]
ChemSpider: n/a PubChem: 17754054

Notes: The carboxy-group was modeled as a charge group on its own. Partial charges for the O and C atoms were chosen as usual (-0.635 on both O and 0.27 on the C) resulting in -1 overall charge.



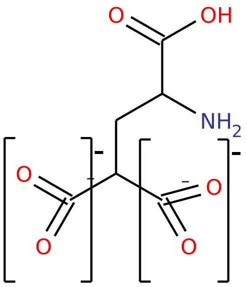
Name: carbamylated Cysteine
Abbreviation: [CAM]
ChemSpider: n/a PubChem: n/a

Notes: Type ga_29 was used for angle SG CD OE1, since SG was considered being chemically close to carbon in this case. As there is no distinct type for SG CD NE2, the same approach was used here. The dihedral CB SG CD NE2 is of type gd_20 to enforce planarity here.



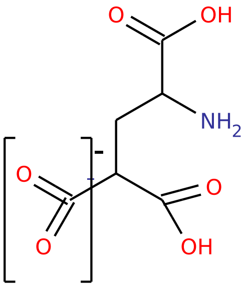
Name: γ-carboxyglutamate
Abbreviation: `[ECA]`
ChemSpider: 37241 PubChem: 40772

Notes: Each of both terminal carboxy-groups has a charge of -1 (resulting in -2 total charge) and they are separated in two groups (2 and 3). An additional improper is required `CG CD1 CD2 CB` compared to canonical `[GLU]`.



Name: γ-carboxyglutamate (protonated)
Abbreviation: `[ECN]`
ChemSpider: 37241 PubChem: 40772

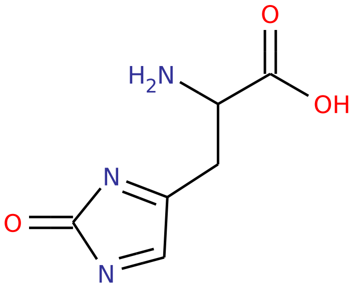
Notes: For the additional H (compared to `[ECA]`), a standard charge has been chosen. Positive charge of atom `CD1` has been increased to match total group charge of 0. The whole terminal rest has charge -1.



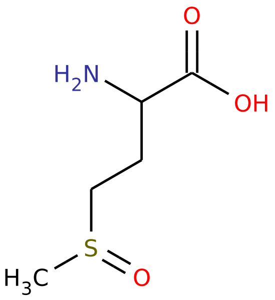
5.4 Oxidations

Name: oxidized histidine
Abbreviation: `[H2X]`
ChemSpider: 19981749 PubChem: 127761

Notes: The standard DNA/RNA bases were used as template: `CG` and `CD2` both get a charge of `0.36`, while the corresponding N atoms have `-0.36`. The remaining carboxy-group can then be modeled in a standard manner.

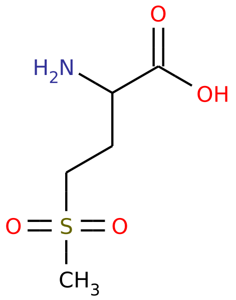


Name: sulfoxide methionine
Abbreviation: [MSX]
ChemSpider: 824 PubChem: 847



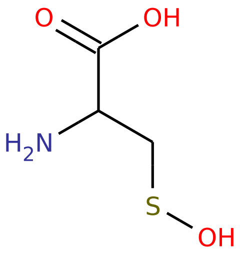
Notes: Charge group 2 has been designed as in [DMSO] with the overall charge is 0. Bond SD CE1 is of type gb_30 because it is longer than gb_29. Bonds of [DMSO] have not been used for consistency reasons. The improper dihedral SD CE1 OE2 CG is enforced by the angles around SD.

Name: sulfone methionine
Abbreviation: [MES]
ChemSpider: 63154 PubChem: 69961



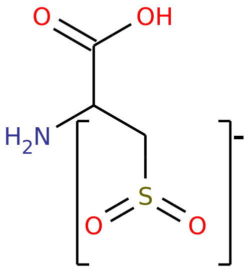
Notes: Same remarks for partial charges as for [MSX]. All angles around SD are of type ga_12.

Name: sulfenic cysteine
Abbreviation: [CYH]
ChemSpider: 144942 PubChem: 165339



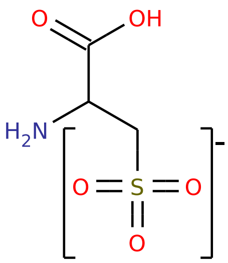
Notes: Atom SG has a charge of 0.15 to match the overall charge of 0. Bond SG OD is type gb_27 since it is significantly longer than the one defined as default in the force field parameters file. The angle and dihedral around OD are as in [SER]: the low energy barrier leads to a rather freely moving H.

Name: cysteine sulfinic acid
Abbreviation: [CSA]
ChemSpider: 107 PubChem: 109



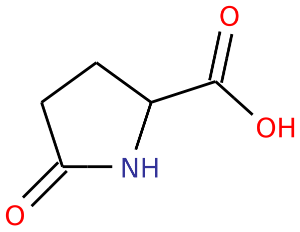
Notes: The positive charge is divided between SG (0.12) and CB (0.15). The overall charge is -1.

Name: cysteic acid
Abbreviation: [CSE]
ChemSpider: 23942 PubChem: 25701



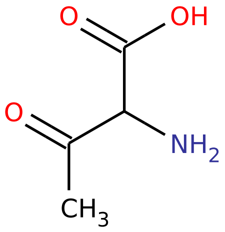
Notes: The charge of the preceding C (CB) is 0.15 as in [CSA]. The high positive charge on SG (0.755) is no problem here since it is surrounded by the O atoms and thus cannot function as a H-bond acceptor anyway.

Name: oxidized proline
Abbreviation: [PGA]
ChemSpider: 7127 PubChem: 7405



Notes: Angles around CD are 1x ga_20 and 2x ga_30 to match 360°. Dihedral CB CG CD N is type gd_20, since CD (flat) is next to CG (non-flat) what results in a multiplicity of six. CG CD N CA has been changed from type gd_19 to gd_4 because of the double bond character.

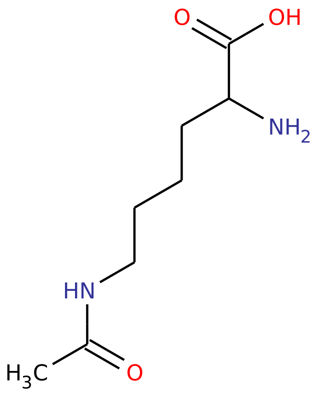
Name: oxidized threonine
Abbreviation: [TOX]
ChemSpider: n/a PubChem: 3014507



Notes: The carboxy group was treated as usual[4.2.3.5], while the type of CB was changed. The three angles around CB are all of type ga_26 to match a total of 360° and sustain planarity in cooperation with improper CB OG1 CG2 CA gi_1.

5.5 Acetylations

Name: N-ϵ-acetyllysine
Abbreviation: [KAC]
ChemSpider: 83801 PubChem: 92832



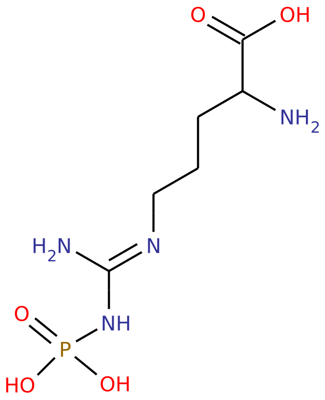
Notes: The segment around the acetyl group has been split up into three different charge groups: atom CI1 has a charge of 0 and the others stick to standard distributions for those small subgroups. Templates for group design were peptide-bond parameters. For reasons of consistency with other modifications, angle CE NZ CH has got type ga_30 here. Atom names are not consistent because of "GROMACS naming problem"[4.2.3.1].

5.6 Phosphorylations

For X-OA-P-X dihedrals, a combination of gd_9 and gd_11 has been applied.

5.6.1 Neutral

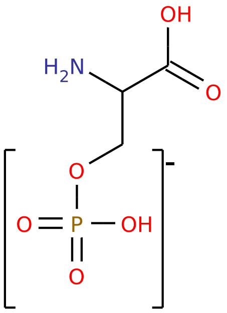
Name: phosphorylated arginine
Abbreviation: [R1P]
ChemSpider: 83195 PubChem: 92150



Notes: The overall charge of the resulting residue is 0, since the contributions of the phospho-group (-1) and the nitrogen-group (+1) cancel out. Although the H of NH2 has no van-der-Waals interactions with the phospho-oxygens, it might interact via electrostatics and therefore exclusions are required.

5.6.2 Charge -1

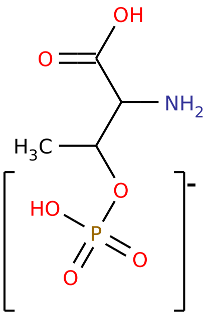
Name: phosphorylated serine
Abbreviation: [S1P]
ChemSpider: 104 PubChem: 68841



Notes: The whole residue from atom CB onwards has been pooled in group 2 to provide a meaningful distribution of the -1 overall charge. Angle ga_27 enforces more planarity than angle ga_13 would (see also [SE2]). Exclusions were added to avoid electrostatic attraction when there are no van-der-Waals interactions[4.2.3.2].

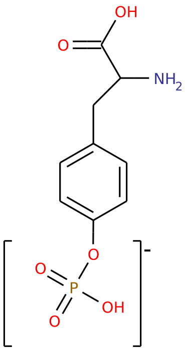
Name: phosphorylated threonine
Abbreviation: [T1P]
ChemSpider: 991 PubChem: 3246323

Notes: Same remarks as for [S1P]. Moreover, an additional improper[4.2.3.4] has been added around atom CB (due to the extra CH3).



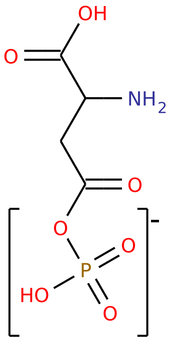
Name: phosphorylated tyrosine
Abbreviation: [Y1P]
ChemSpider: 28593 PubChem: 30819

Notes: Same remarks as for [S1P]. Extra restraints are required between atom HI3 and the three additional oxygens.



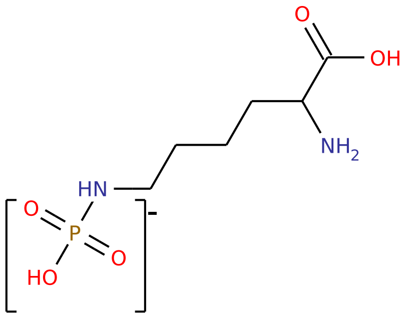
Name: phosphorylated aspartate
Abbreviation: [D1P]
ChemSpider: 134354 PubChem: 152441

Notes: The whole terminal section, including CG and OD1 were combined in one charge group. The charge of OD1 was changed to -0.38, which is often used for the O in keto-groups[4.2.3.5] and CG was used to tune the overall charge to -1. Finally, in contrast to other phosphorylations, dihedral gd_3 was used for CB CG OD2 PE (because of atom CG).



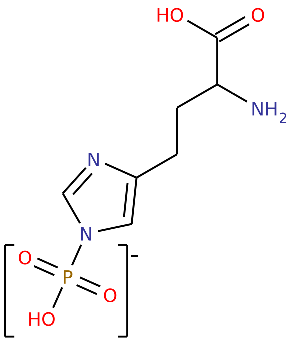
Name: 6'-phosphorylated lysine
Abbreviation: [K1P]
ChemSpider: 141512 PubChem: 161086

Notes: Type of NZ changed to NE, because of its collaboration with the rest of the system. For bond NZ PH, gb_23 fits best since it is shorter than gb_24 and thus accounts better for double bond character (see [K2P]). The angles around NZ sum up to 360°. Partial charge of PH was changed because of the split up of the charge groups.



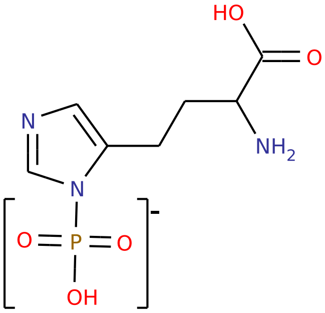
Name: 1'-phosphorylated histidine
Abbreviation: `[H1P]`
ChemSpider: 4367287 PubChem: 15458486

Notes: For angles, dihedrals and charge groups, see `[H31]`. The phospho-group is attached to atom `NE2`, which has no partial charge since it has been modeled independently.



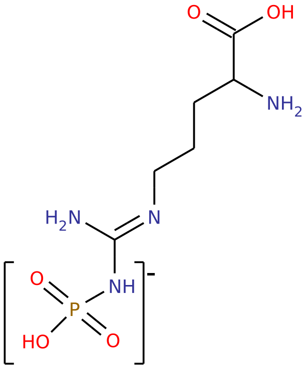
Name: 3'-phosphorylated histidine
Abbreviation: `[H31]`
ChemSpider: 19980564 PubChem: 17754026

Notes: The planar ring structure (total charge 0) and the phospho-group (total charge -1) have been separated into distinct charge groups to gain consistency with regard to other phosphorylations. The dihedral `CG ND1 PE3 OZ3` is a sole `gd_19` to enable free rotation.



Name: phosphorylated arginine
Abbreviation: `[R2P]`
ChemSpider: 83195 PubChem: 92150

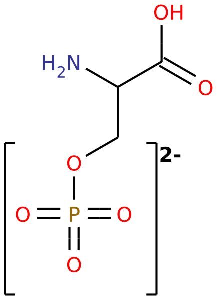
Notes: The phospho-group has a charge of -2, distributed only over `PT` and the oxygen atoms. To match the overall charge, it was necessary to set `PT` to `-0.095`. This has no impact on its ability to act as an H-bond acceptor since it is occluded anyway and is thus blocked.



5.6.3 Charge -2

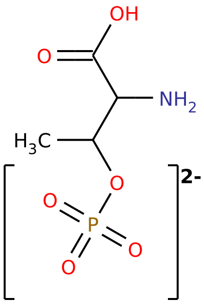
Name: phosphorylated serine
Abbreviation: [S2P]
ChemSpider: 104 PubChem: 68841

Notes: All angles around atom PD are of type ga_13 in order to maintain the whole group in the tetrahedral shape. This seems needed because of the lone electron pair of PD (compared to [S1P]). All three terminal O atoms are of type OM.



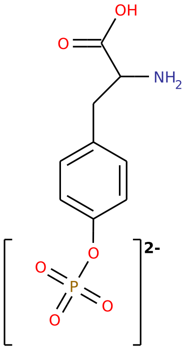
Name: phosphorylated threonine
Abbreviation: [T2P]
ChemSpider: 991 PubChem: 3246323

Notes: Same remarks as for [T1P], besides OE3 has been changed to type OM.



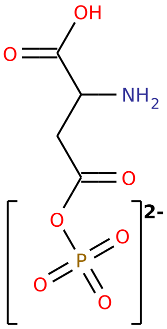
Name: phosphorylated tyrosine
Abbreviation: [Y2P]
ChemSpider: 28593 PubChem: 30819

Notes: Same remarks as for [Y1P]. Exclusions, bonds, angles and dihedrals adapted due to deletion of atom HI3 compared to [Y1P].



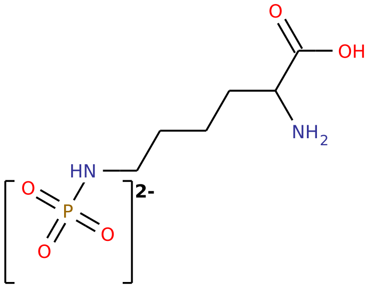
Name: phosphorylated aspartate
Abbreviation: [D2P]
ChemSpider: 134354 PubChem: 152441

Notes: Same remarks as for [D1P]. Charges of CG and OD1 stay the same, the phospho-group has been implemented as in [S2P] (except bond CG OD2 and dihedral CB CG OD2 PE, because of the divergent type of CG).



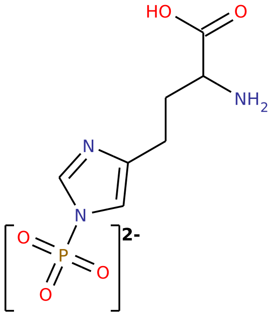
Name: 6'-phosphorylated lysine
Abbreviation: [K2P]
ChemSpider: 141512 PubChem: 161086

Notes: As in [K1P], the dihedrals CD CE NZ PH and CE NZ PH OT3 are of type gd_19 (no additional gd_11). The first one is from arginine (NZ is of type NE), the second ensures a low energy barrier leading to almost free rotation and a multiplicity of six. Again, the charge groups have been split up, leading to a negative charge on atom PH.



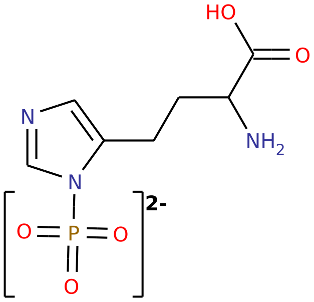
Name: 1'-phosphorylated histidine
Abbreviation: [H12]
ChemSpider: 4367287 PubChem: 15458486

Notes: Same considerations for the charge groups as in [R2P]. The partial charge distribution of the ring has been located at and around the unbound nitrogen: -0.58 at ND1 and 0.29 at each of the surrounding C atoms.

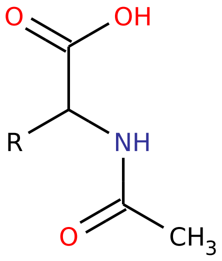


Name: 3'-phosphorylated histidine
Abbreviation: [H32]
ChemSpider: 19980564 PubChem: 17754026

Notes: Same remarks as for [H12] in regard of the charge group arrangement.

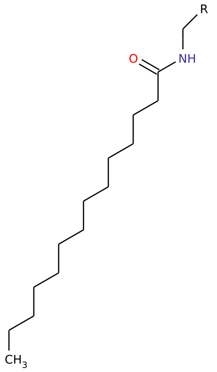


5.7 N-terminal modifications



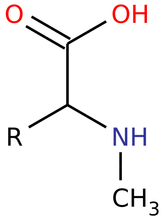
Name: acetylation
Abbreviation: [NAC]
ChemSpider: n/a PubChem: n/a

Notes: Alteration of charge groups is required after addition of atoms CN1, ON2 and CN2. The latter one is separated with charge 0.0, while the others have been modeled as standard carboxy-group. Additional impropers around N and CN1 have been specified.



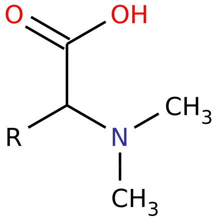
Name: myristoylation
Abbreviation: [MYR]
ChemSpider: n/a PubChem: n/a

Notes: The whole terminal alkyl group is in one charge group. All members which are united C atoms got a partial charge 0.0, the carboxy function has been modeled as usual.



Name: N-methylation
Abbreviation: [1NM]
ChemSpider: n/a PubChem: n/a

Notes: United C atom CN1 is added to N and the charge group is extended to include CA as well. The positive charge is split up into H1 (0.41), CA and CN1 (each 0.21).

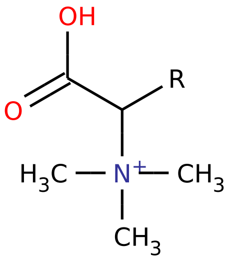


Name: N,N-dimethylation
Abbreviation: [2NM]
ChemSpider: n/a PubChem: n/a

Notes: Two methyl groups are added, each with a charge contribution of 0.21 which sums up to 0.63 together with CA.

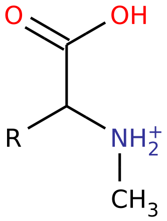
Name: N,N,N-trimethylation
Abbreviation: [3NM]
ChemSpider: n/a PubChem: n/a

Notes: The total N-terminal charge is +1. It is evenly distributed over the three additional methyl groups, CA and the nitrogen.



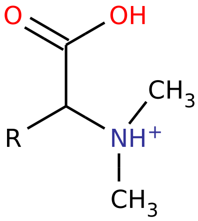
Name: N-methylation (charged)
Abbreviation: [1NC]
ChemSpider: n/a PubChem: n/a

Notes: One methyl group (CN1) and one hydrogen (H2) are added and atom H is renamed to H1. The total charge of the N-terminus is +1.



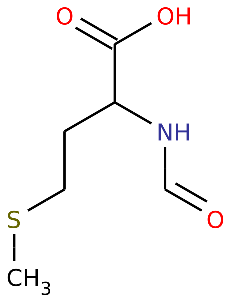
Name: N,N-dimethylation (charged)
Abbreviation: [2NC]
ChemSpider: n/a PubChem: n/a

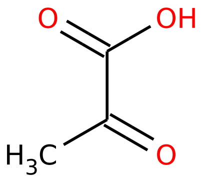
Notes: Total charge of +1 is distributed over the five N-terminal atoms.



Name: N-formylmethionine
Abbreviation: [FOM]
ChemSpider: 887 PubChem: 25244433

Notes: All added atoms (CN1, ON2 and H1) are regrouped in the first charge group and modeled as an aldehyde.

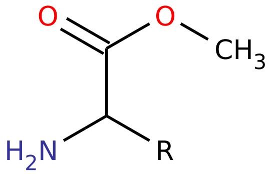




Name: pyruvic acid
Abbreviation: [PYA]
ChemSpider: 1031 PubChem: 1060

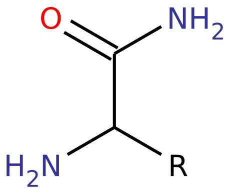
Notes: The nitrogen is removed and atoms CB1 and OB2 are added. Bond CA C is of type gb_22, which is short and thus accounts better for the introduced double bond character.

5.8 C-terminal modifications



Name: methylation
Abbreviation: [CME]
ChemSpider: n/a PubChem: n/a

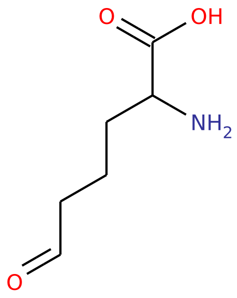
Notes: Type of one oxygen is changed to OA. Atom OC1 has standard carboxy partial charge, while the remaining atoms are assigned as in [EME].



Name: amidation
Abbreviation: [AMD]
ChemSpider: n/a PubChem: n/a

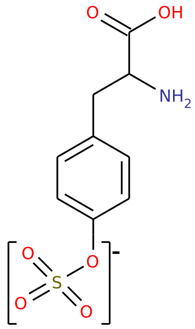
Notes: The C-terminal amino group is modeled separately and thus has standard partial charge values.

5.9 Miscellaneous modifications



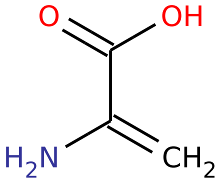
Name: allysine
Abbreviation: [KAL]
ChemSpider: 202 PubChem: 160603

Notes: The aim was to get a strong orientation at the terminal group (dipole) and to keep the positive charge of the C atom low. Therefore, atom CE was modeled as bare carbon with explicit hydrogen, which allows charge distribution and an additional improper.



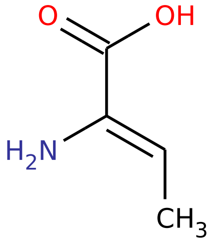
Name: sulfotyrosine
Abbreviation: [YSU]
ChemSpider: n/a PubChem: n/a

Notes: From atom CZ onwards, the whole terminal end has been included in one charge group (number 7). For consistency reasons, charges on the O atoms are standard and ST has been used to adjust the overall charge[4.2.3.5]. For explanation with regard to dihedrals, see building block [PGA].



Name: dehydrated serine
Abbreviation: [SDH]
ChemSpider: n/a PubChem: n/a

Notes: The covalent connection from atom CA to atom CB has single bond character in the native template. Thus, bond type ga_9 was used in this case which is significantly shorter: 1.33 compared to 1.53 Å.

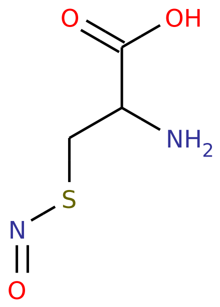


Name: dehydrated threonine
Abbreviation: [TDH]
ChemSpider: n/a PubChem: n/a

Notes: Same remarks as for [SDH].

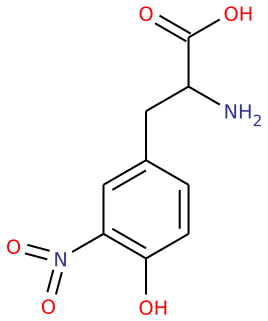
Name: S-nitrocysteine
Abbreviation: [CSN]
ChemSpider: n/a PubChem: 3359494

Notes: Charge of atom ND is 0.28 because of delocalization effects. This also results in a [MET]-like bond between SG and ND. The bond between SG and OE has double bond character. Dihedral CB SG ND OE has been modeled as if it were X-S-S-X (high barrier, muliplicity of two).



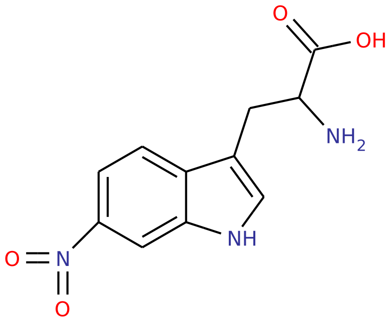
Name: nitrotyrosine
Abbreviation: [YNI]
ChemSpider: 205676 PubChem: 65124

Notes: The negative charges of the oxygens and atom CE1 result in a positive one on NZ1 (0.86), the overall charge is 0. Improper NZ1 CE1 OH1 OH2 accounts for the interaction of the nitro group with the ring structure which leads to planarity. Bond CE1 NZ1 is of type gb_11 since NZ1 is type NR.



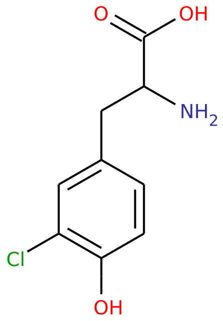
Name: nitrotryptophan
Abbreviation: [WNI]
ChemSpider: 521768 PubChem: n/a

Notes: Same remarks as for [NTY].



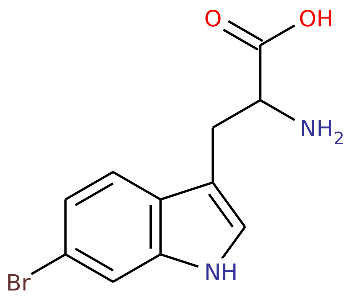
Name: chlorotyrosine
Abbreviation: [YCH]
ChemSpider: 106510 PubChem: 119226

Notes: The small charge distribution chosen between atom types C and CL as well as the selected bond (gb_37) are standard values for this kind of construct. In terms of angles, atom CLZ1 has been treated as a non-hydrogen 6-ring member.



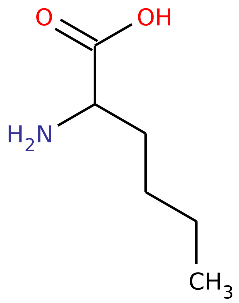
Name: bromotryptophan
Abbreviation: [WBR]
ChemSpider: 2736705 PubChem: n/a

Notes: Partial charges are derived from electronegativity in halogens: For atoms BRT and CH2 values of -0.055 and 0.055 have been chosen.



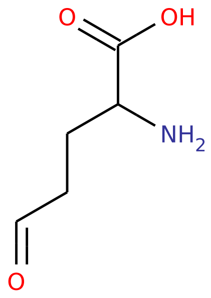
Name: norleucine
Abbreviation: [NLO]
ChemSpider: 9103 PubChem: 9475

Notes: The side-chain has no partial charges. Angles between united C atoms CA to CE are of type ga_14.



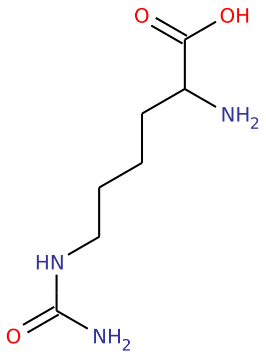
Name: glutamic semialdehyde
Abbreviation: [GSA]
ChemSpider: 167744 PubChem: 193305

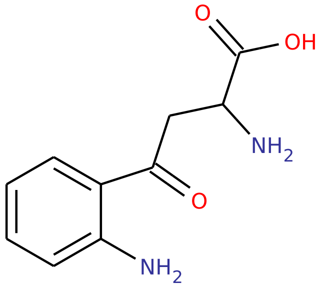
Notes: The charge of atom OE is that of the backbone oxygen (-0.38) and the positive charge has been distributed over atoms CD and HD to approximate an aldehyde function.



Name: citrulline
Abbreviation: [RCI]
ChemSpider: 9367 PubChem: 9750

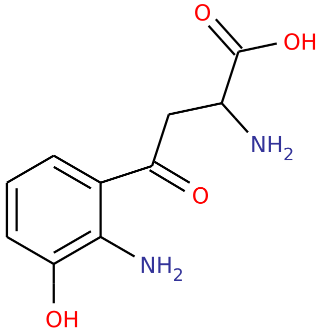
Notes: The terminal part has been split up in 3 distinct charge groups, where CZ and OH1 build up the middle part.





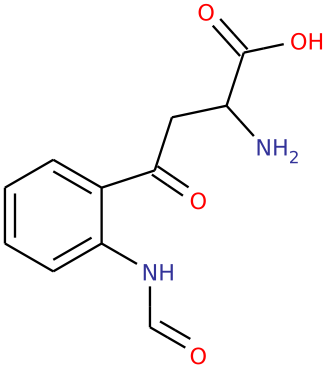
Name: kynurenine
Abbreviation: `[WKY]`
ChemSpider: 823 PubChem: 846

Notes: Two C atoms are removed from the pyrrole ring of tryptophan and oxygen `OD1` is added to atom `CG`. All charge groups are standard and since the benzene is not affected, it remains aromatic.



Name: hydroxykynurenine
Abbreviation: `[WKH]`
ChemSpider: 87 PubChem: 89

Notes: Same remarks as for `[WKY]`. Of the benzene ring, only atom `CZ2` is affected by the addition of a hydroxy group.

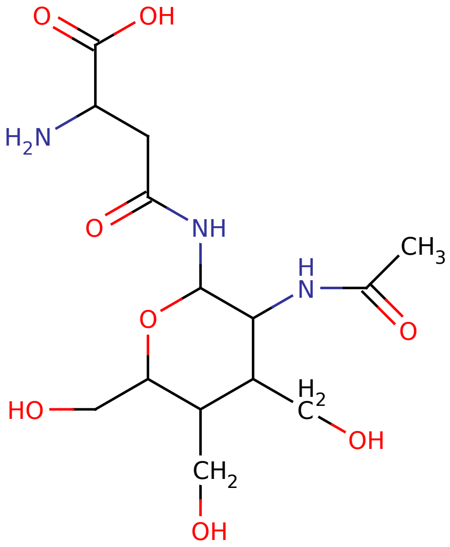


Name: formylkynurenine
Abbreviation: `[WKF]`
ChemSpider: 886 PubChem: 910

Notes: Same remarks as for `[WKY]`. The additional aldehyde group is considered separately from `NZ1`. Dihedrals are of type `gd_2`, because it is more flexible than the peptide bond (lower energy barrier).

Name: N-glycosylated asparagine
Abbreviation: [NNG]
ChemSpider: n/a PubChem: n/a

Notes: The type of atom ND2 is changed from NT to N (same type as in peptide backbone). Atom HD21 is renamed to HD2 and the second hydrogen is deleted. As for charge groups, the sugar ring structure is split up into smaller subgroups. Note that the atom naming scheme for this modification does not follow the rules stated in subsection 4.2.3.1. For X-CHx-N-X dihedral angles type gd_19 has been used.



5.10 Differences between ffG45a3 and ffG54a7

For most modifications, the same parameterization strategy can be used for both GROMOS force field versions considered. The remaining changes are listed below.

- In ffG54a7, an additional oxygen type has been introduced which belongs to carboxy groups (without hydrogens). Therefore, this atom type is used in ester groups (example: `[EME]`).
- Compared to ffG45a3, in ffG54a7 the sp^3 to sp^2 shift (e.g. in `[EME]`) is also included: `ga_22` has been used for angle `CD OE2 CZ`.
- If the same procedure as in ffG45a3 would have been applied for residues as `[H1M]`, the partial charge on the methyl group would have been too big. Thus, it has been split up.
- For cases where a -PO_3 group is added directly to a nitrogen atom (e.g. `[H11]`), the nitrogen has been excluded from the charge group. Its charge was set to `0.0`, while the carbon atom charge was increased to reduce it on the hydrogens.
- In all phosphorylated residues where the phospho group is connected via an oxygen, the type of this O atom was changed to OE since it has no hydrogen attached any longer. Therefore, this is necessary to avoid hydrogen bonds there.
- In general, since there are more available parameters in the new force field, the identifiers of the corresponding bonds, angles *et cetera* were shifted although they might be of the same value in both versions.

CHAPTER 6

Definitions

In this section all file formats, which are used by the server, are defined and explained. All of them, except the instruction file which is generated on-the-fly for each particular job, can be edited directly via the admin panel[4.1.1.7]. Note that changes of these files have server-wide effects and thus should be made with caution.

6.1 Self-defined formats

The following formats have been designed to serve as customized data containers for the workflow. The aim was to implement them to be both intuitive and efficient in terms of parsing. In all cases, there is no defined order of statements within blocks. Some statements are mandatory, while others are only required in distinct cases (see details below). A sophisticated error reporting system is only available for the instruction file.

6.1.1 Instruction file

The communication between the frontend and the backend is mainly facilitated by passing the instruction file. It is generated by the frontend for each distinct job and contains all the options specified by the user.

Listing 6.1: Instruction file example

```
EXECTIMELIMIT 60
FFTYPE ffG45a3
COPYHEADER DISABLED
DEBUMPFLAG DISABLED
MINIFLAG ENABLED
RESTRAIN ENABLED 10000
MOD RESNO=45 CHAINID=A TYPE=METY
MOD RESNO=50 CHAINID=A TYPE=HYDR
MOD RESNO=34 CHAINID=B TYPE=HYD3
MOD RESNO=84 CHAINID=B TYPE=PHOS
MOD RESNO=77 CHAINID=A TYPE=PHOS
MOD RESNO=24 CHAINID=D TYPE=PHO2
MOD RESNO=99 CHAINID=D TYPE=KYNU
```

There is no maximal number of modifications which can be applied in one job[1]. To run a large series of distinct jobs, it is sufficient to establish a script taking control of instruction file and folder

[1] Note, that a vast number of modifications may lead to abortion of minimization because of the time limit. This is also dependent on the molecule's size.

structure generation as well as the backend call. There is no intrinsic order among the MOD statements since selections might be done without regularity.

Details:

- EXECTIMELIMIT expects an integer number, which represents the seconds a job is allowed to take before it is aborted. This is particularly useful to limit both the server load and the user's maximum waiting time. Can be set for all jobs using the administration panel.
- FFTYPE defines the selected force field version, which is used afterwards during file manipulation and minimization. It also specifies the minimum bond lengths used for addition of atoms during modification. Values which are currently supported: ffG45a3 and ffG54a7.
- COPYHEADER (flag) can be either ENABLED or DISABLED and allows copying of the header region of the initial PDB file to the final one. The header is **not** updated automatically, so it might be corrupted in some cases. Therefore, the default value is DISABLED.
- DEBUMPFLAG (flag) allows additional debumping in case minimization is not sufficient to resolve severe clashes. It is recommended to try minimization first.
- MINIFLAG (flag) enables or disables minimization after modification. If minimzation is selected, but the specified PDB file does not pass the initial validation check, this flag is automatically set to DISABLED in order to avoid a useless try after modification[2].
- RESTRAIN (flag) is set to one of the following: DISABLED, ENABLED or BACKBONE. The value ENABLED implements position restraining to all atoms belonging to non-modified residues, while the latter restrains only the backbone atoms. The second argument specifies how strong the restraints will be (in all directions).
- MOD statements define the required modifications for this job. At this stage, it does not matter which type of amino acid is modified[3] since residue dependent differences are encoded in the modifications database.
  - RESNO is the number of the particular residue[4].
  - CHAINID states the corresponding chain-identifier in case there is one given.
  - TYPE defines the four-letter abbreviations used to distinguish classes of modifications (see also table 9.2).

These files are named PDBNAME.pdb.cfg and have to be placed in the job's execution directory together with PDBNAME.pdb. All the files generated subsequently by the workflow will also be located in the very same folder.

[2] The user is informed in advance of the selection step.

[3] As long as this modification is available for a given amino acid in the defined force field.

[4] The one defined in the PDB file, which may be different from the real count due to missing residues.

6.1.2 Modifications database file

Each force field type and version requires its own modification database file, which specifies in what manner atoms have to be added or deleted in order to result in the desired modifications. The structure of those files is as follows. Each type of modification (e.g. -1 charge-phosphorylation) is represented as a block. The starting point is indicated by a directive `[MOD=????]` (the four-letter code has to match with table 9.2) and finalized by `[ENDMOD]`. Within these blocks there is an entry fo each canonical residue for which this particular modification is available[5]. Residue sub-blocks contain all the information needed for alteration, encoded in a variety of statements. Their order is arbitrary, but it is mandatory that each has to be exclusively on its own separate line. Statements and their related values are either separated by a blankspace or an equal sign in case of the `ADD` directive.

Listing 6.2: Excerpt of modifications database for ffG45a3

```
// PHOSPHORYLATION (-1 CHARGE)
[MOD=PHOS]
[RES=ARG]
NTP P0R
DVE CZ NH2
ANC NH2
ADD NAME=PT TYPE=P XCOORR=0 YCOORR=0 ZCOORR=1.610 TEMPF=0
ADD NAME=OI1 TYPE=O XCOORR=0 YCOORR=1.610 ZCOORR=1.610 TEMPF=0
ADD NAME=OI2 TYPE=O XCOORR=0 YCOORR=0 ZCOORR=3.220 TEMPF=0
ADD NAME=OI3 TYPE=O XCOORR=0 YCOORR=-1.480 ZCOORR=1.610 TEMPF=0
ADD NAME=HI3 TYPE=H XCOORR=0 YCOORR=-1.480 ZCOORR=2.610 TEMPF=0
REP HH21 HH2
DEL HH22
[ENDRES]
[ENDMOD]
```

Details:

- `NTP`: New amino acid identifier consisting of three letters[6]. Note: In the original rtp file, there are also compounds defined, which carry names with four or even more letters. However, these are not consistent with the current PDB file format.

- `DVE`: Vector built up by two atoms of the original sidechain. In the majority of cases, it is the last remaining bond. The second argument is the tip of the vector and its direction is that of relative axis `Z` (see also 4.1.2.2).

- `ANC`: Anchor point which serves as the origin of the relative coordinate system. Can be any atom of the original residue, which is not deleted by `DEL` statements, but usually its the one atoms are attached to. It has to be specified even if addition is not required (in this case it has no effect).

- `DEL`: Deletes an atom from the canonical amino acid.

- `REP`: Replaces the name of the atom specified by the first argument with the value of the second argument. Does not affect atom positions. In some cases, it is more straightforward to simply delete an atom and add a new one instead of using `REP` since the former procedure also allows to redefine position.

[5] Amino acids are identified by their standard three letter code.

[6] Numbers are also allowed.

- NGR and CGR: Requires an integer value as input. Specifies “N-terminal GRoup” (and “C-terminal GRoup” respectively) which are required for terminal modifications. Those are applied by pdb2gmx, which does not support setting these options by flags. Therefore, the numbers of the menu entries have to be defined (see table 9.1 for details) to call them during execution.

- CCC: Sets charge group numbers in case they have to be changed (this flag concerns terminal modifications[4.1.2.4] only).
  - NAME: The name of the atom in the topology file whose charge groups have to be altered.
  - TYPE: Atom type (element) of the particular atom. Also includes united atoms.
  - NEWGROUP: For N-terminal modifications: A positive integer number (at least 1) is given, which is assigned as the new charge group (e.g. 3). For C-terminal modifications: A relative shift is specified, since atoms are added at the end (e.g. +2). The unaffected atoms of the same residue as well as the ones following are changed accordingly.
  - NORC: Flag specifying if the modification is N- or C-terminal, since those two differ in implementation. Values are either NTERM or CTERM.

- ADD: Adds a new atom to the residue.
  - NAME: The name of the atom (has to be consistent with the specifications).
  - TYPE: Element of the atom. Includes united atoms as well.
  - XCOORR, YCOORR, ZCOORR: Relative coordinates with regard to the anchor point. The first axis which should be used is Z, since it usually points outwards.
  - TEMPF: Temperature factor, which is 0 since the atom’s position was not determined experimentally.

6.1.3 File for program calls

The backend needs to call a series of programs to render minimization and application of terminal modifications possible. There is one file for specification of minimization steps and two files for debumping.

Listing 6.3: Excerpt of minimization protocol

```
<STEP>
<FINFILE>output.gro
<COMMAND>/usr/local/gromacs/bin/pdb2gmx -f [RELATIVE_PATH]mod_[PDB_FILENAME]
-o [RELATIVE_PATH]output.gro -p [RELATIVE_PATH]output.top -ignh -ter -missing
-ff G45a3 -i [RELATIVE_PATH]posre.itp
<NTERMINUSMOD> true
<CTERMINUSMOD> true
</STEP>

<STEP>
<FINFILE>output.tpr
<COMMAND>/usr/local/gromacs/bin/grompp -v -f minimize_ffG45a3.mdp
-c [RELATIVE_PATH]output.gro -p [RELATIVE_PATH]output.top
-o [RELATIVE_PATH]output.tpr -po [RELATIVE_PATH]mdout.mdp -maxwarn 1
<RESTRAINTSSTEP>true
</STEP>
```

New lines are just at the ends of the statements and not in between. They are parsed and executed one after the other starting at the top.

Details:

- `<STEP>` and `</STEP>` incorporate one command, which means they are used to define a consistent block of directives.
- `<FINFILE>` is the file, which is used to determine whether the current step has been completed or not. This is important to ensure that all input files required in the next step are already available.
- `<COMMAND>` specifies the program call, which is made from the server's home directory.
  - `[RELATIVE_PATH]` is replaced by the job's path as seen from the home directory.
  - `[PDB_FILENAME]` is replaced by the original file name (mind the prefix).
- `<NTERMINUSMOD>` and `<CTERMINUSMOD>` are flags used to mark steps, which are required for appliance of terminal modifications.
- `<RESTRAINTSSTEP>` (flag) is used to mark steps which are suspended until the manipulation of file `posres.itp` is completed in order to apply restraints.

6.1.4 Logfile

The files `logfile.log` and `status.log` contain all major events and statistics with regard to job processing. They contain the following information:

- whether input files could be loaded successfully.
- which modifications could be applied and (eventually) which problems appeared.
- which output files could be written.
- whether the job was successful or failed completely / partially.
- the elapsed time.
- number of deleted and added atoms.

6.2 Various formats

6.2.1 PDB file format

The PDB file format[7] has been specified by the **w***orld***w***ide ProteinDataBank* [62]. A PDB file includes structural information about one or more molecules (mostly proteins and nucleic acids). The name of the file (the identifier) is a four letter code, which uniquely defines the molecule within. Its native extension is `*.pdb` but since it is a plain text-file, this is not mandatory for most programs. A proper file begins with a header region[8] in which data about the following content is stored. For instance, there are entries describing experimental conditions, authors, disulfide bonds and many more. Every line has a keyword at the beginning, which has a maximal size of 6 letters and the line as a whole a length of 80. Different information blocks in a line are discerned only by their positions. Obviously, this simplicity involves intolerance against any deviation from the norm. Thus it is crucial to check PDB files for validity before using the workflow.

The main information of a PDB file, i.e. atom coordinates, is encoded in a list of `ATOM` (canonical residues) and `HETATM` (non-canonical residues) statements. As long as the character range for a given block is not exceeded, it does not matter whether blankspaces are used to fill up a block or not.

Listing 6.4: Example for ATOM records

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
--------------------------------------------------------------------------------

...
ATOM     43  C   ASP A  61     -19.210 107.745 -85.387  1.00 81.28           C
ATOM     44  O   ASP A  61     -18.291 108.336 -85.944  1.00 81.37           O
ATOM     45  CB  ASP A  61     -19.389 107.736 -82.877  1.00 83.63           C
...
```

Keyword ⟷ Atomname ⟷ ⟵ Coordinates ⟶

A full description of all sections can be found here: [62].

[7] Version considered in this work is 3.20 (published in 2008).

[8] Not to be mistaken with the `HEADER` line, consisting of classification, the generation date and the PDB identifier.

6.3 GROMACS files

The content of GROMACS files is grouped in blocks with a header directive: `[ directive ]`. Everything after a ”;“ (semicolon) is ignored by the various binaries and used for commenting. It is possible to use a C++ preprocessor-like syntax to include other files: `#include "FILE"`. Moreover, constant variables can be defined using `#define`: `#define gb_1   0.1000 1.5700e+07`.

6.3.1 Topology: TOP file

Text-file containing topology information about frequently used molecules[9] such as water as well as metadata and all atoms, bonds, angles and dihedrals in the considered molecule[10]. It is therefore also called ”systems topology“ in contrast to the `*.itp` file included within.

Listing 6.5: Example for GROMACS top file

```
[ atoms ]
; nr type resnr residue atom cgnr charge mass
     1         NL     55    GLY      N      1      0.129    14.0067
     2          H     55    GLY     H1      1      0.248      1.008
...

[ bonds ]
; ai aj funct
    1     2     2    gb_2
    1     3     2    gb_2
...

[ pairs ]
; ai aj funct
    1     7     1
    1     8     1
...

[ angles ]
; ai aj ak funct
    2     1     3     2    ga_9
    2     1     4     2    ga_9
...

[ dihedrals ]
; ai aj ak al funct
    1     5     6     8     1    gd_20
    5     6     8    10     1    gd_4
...
```

[9] Included in the header region, see subsection 6.3.3.

[10] Version of GROMACS manual used: 4.5.4 (published in 2010).

6.3.2 C- and N-terminal files

In the case of a non-standard N- or C-terminus, `pdb2gmx` has the possibility to select different termini which are encoded in `ffG45a3-c.tdb` and `ffG45a3-c.tdb`, respectively. To enable this selection, argument `-ter` has to be set when calling the binary.

Listing 6.6: Excerpt of extended N-terminal modification file

```
[ NAC ]
[ add ]
1       2       H       N       C       CA
        H       1.008   0.2800
1       1       CN1     N       C       CA
        C       12.011  0.3800
1       2       ON2     N       CA      C
        O       15.9994 -0.3800
1       2       CN2     N       C       CA
        CH3     15.035  0.0000
[ bonds ]
N       H       gb_2
N       CN1     gb_9
CN1     ON2     gb_4
CN1     CN2     gb_26
[ angles ]
CA      N       H       ga_17
CN1     N       H       ga_31
CN1     N       CA      ga_30
N       CN1     ON2     ga_32
N       CN1     CN2     ga_18
CN2     CN1     ON2     ga_29
[ impropers ]
N       CN1     CA      H       gi_1
CN1     CN2     N       ON2     gi_1
[ dihedrals ]
CN2     CN1     N       CA      gd_4
CN1     N       CA      C       gd_19
```

Note that the name of the block (e.g. `NAC` in this case) is not the name of the residue afterwards. A terminal amino acid which is modified, retains its name in contrast to a building block exchange within a peptide. It is also possible, to enable certain modifications only for a defined subset of amino acids.

Caution: A change of or an addition to an existing list of terminal modifications may alter their position and thus their indices even if an entry is made at the very end of the file. This is due to the way `pdb2gmx` orders the menu which is designed to be used by a human being rather than being fed by an automatized workflow. It is therefore highly recommended to check how the menu has changed and to update the corresponding numbers in the modification database files.

6.3.3 Topology: ITP file

These "include topology" files define molecules, which are widely used (e.g. water) and can be included as reusable blocks in topology files. They do not have:

- `#include` statements
- the following directives: `[molecule]` and `[system]`

A full description of all sections can be found here: [8].

6.3.4 Building blocks: RTP file

The information as to how chemical compounds in general and (for this workflow) amino acids in particular are modeled is stored in `*.rtp` files. Dependent on the version used these are either named `aminoacids.rtp`[11] or `ffG45a3.rtp`.

Listing 6.7: Example for GROMACS rtp file

```
[ CYM ]
 [ atoms ]
    N     N    -0.28000     0
    H     H     0.28000     0
   CA   CH1     0.00000     1
   CB   CH2     0.00000     1
   SG     S     0.00000     2
   CD   CH3     0.00000     2
    C     C       0.380     3
    O     O      -0.380     3
 [ bonds ]
    N     H    gb_2
    N    CA    gb_20
   CA     C    gb_26
    C     O    gb_4
    C    +N    gb_9
   CA    CB    gb_26
   CB    SG    gb_30
   SG    CD    gb_29
 [ angles ]
; ai aj ak gromos type
   -C     N     H      ga_31
    H     N    CA      ga_17
   -C     N    CA      ga_30
    N    CA     C      ga_12
   CA     C    +N      ga_18
   CA     C     O      ga_29
    O     C    +N      ga_32
    N    CA    CB      ga_12
    C    CA    CB      ga_12
   CA    CB    SG      ga_15
   CB    SG    CD      ga_3
 [ impropers ]
; ai aj ak al gromos type
    N    -C    CA     H      gi_1
    C    CA    +N     O      gi_1
   CA     N     C    CB      gi_2
 [ dihedrals ]
; ai aj ak al gromos type
  -CA    -C     N    CA      gd_4
   -C     N    CA     C      gd_19
    N    CA     C    +N      gd_20
    N    CA    CB    SG      gd_17
   CA    CB    SG    CD      gd_13
```

[11] In this case, force field specification is facilitated by the folder path.

These files contain:

- An initial block identifier (e.g. CYM in this case), which is the residue's name (and therefore has to be consistent with the one in PDB files).
- [atoms]: The first entry is the atom name (same as in the PDB file) and the second one specifies its type (9.4.1). The third argument is the partial charge and the last one the charge group it belongs to. The total charge of a particular group should be exactly 0 except that it carries a defined charge. It is recommended to state one charge group after the other.
- [bonds]: Covalent connections between atoms (9.4.2).
- [angles]: Angles defined by three atoms which are linearly connected (9.4.2).
- [exclusions]: In case more than the usual exclusions for non-bonded interactions (between atoms separated by three or less bonds) are required because of the spatial structure, they can be stated here.
- [impropers]: Used to ensure planarity or a distinct stereo-chemistry (9.4.2).
- [dihedrals]: Dihedral angles require 4 connected atoms and describe the angle between two planes defined by these atoms (9.4.2).

A full description can be found in the GROMACS manual: [8]. The theoretical backgrounds of the parameters and the potentials they belong to are further discussed in subsection 4.2.2.

6.3.5 Residuetypes file

This file (aminoacids.dat) is required as a list of supported building blocks. The first line contains the number of entries below, each one on a separate line. Moreover, it is noteworthy that this file is different in the new GROMACS versions (4.5.x and up) where also the type of macromolecule a residue belongs to is stated[12].

Listing 6.8: Excerpt of extended aminoacids.dat

```
241
1NC
1NM
2NC
2NM
3NC
F23
W2H
F2H
F3H
W4H
W5H
W6H
W7H
WNI
RMA
ABU
ACE
AIB
...
```

[12] Separated by blankspaces, e.g. 2NM Protein.

6.3.6 Hydrogen database

If hydrogen atoms are missing, they can be automatically added by `pdb2gmx` if there is an appropriate entry in the file `ffG45a3.hdb`.

Listing 6.9: Excerpt of extended `ffG45a3.hdb`

```
SE1     2
1       1       H       N       -C      CA
1       2       HE3     OE3     PD      OG
```

The first line contains the name of the building block and the number of additions (in lines) which are supposed to follow. The next lines define how many hydrogen atoms are added, what type of method is used, the name of the atom and three or four control atoms (the first one is the atom to which the hydrogens are attached to, while the others depend on the method).

CHAPTER 7

Conclusion

In the present work, Vienna-PTM, a server to simplify the *in silico* treatment of protein post-translational modifications, has been introduced. The workflow allows to upload PDB files (or files specified by their PDB identifier) and to select the desired modifications by an intuitive menu. These alterations are applied afterwards in a fully automatic way, ensuring consistency and thus limiting errors to a minimum, while providing high-speed modification. In conjunction with the available force field parameter packages, Vienna-PTM offers support for all steps needed to run molecular dynamics simulations with modified proteins.

As for this thesis, the server's structure and functionality as well as the extended amino acid alphabet and its biological relevance have been presented in detail. As pointed out, the computational description of PTMs is not just a minor addition to force field parameters, but rather renders an important layer of macromolecule diversity in cells accessible to computational modeling. Possible areas of application are e.g. studies on stability, dynamics and interactions of proteins which can be simulated with different types and levels of modifications. Moreover, the workflow might help to shed light on the fundamental principles underlying PTM-related effects on proteins in general.

Currently, 219 different building blocks are supported, of which 150 are unique while the rest represents different charge states or stereochemistry versions, for GROMOS force fields G45a3 and G54a7. Modified residues are treated as "new" amino acids, in the sense that they have unique names and are placed in the ATOM section of PDB files. If atoms have to be added, this is facilitated by using pre-minimized coordinates, in order to guarantee meaningful initial positions in any case. However, to get a structure taking the whole environment into account, subsequent minimization is necessary. Note that because of the ongoing maintenance of the server and the parameter files, this thesis might contain inconsistencies of different level at a later time point.

Outlook

Because of the modular design of the server, with all key parameters being listed in plain text files, it is straigthforward to add new modifications without changing the core modules, namely the server-side scripts and the backend. Furthermore, even the implementation of new force fields is supported, including required program calls and process monitoring, since those specifications are out-sourced and thus extendible too. In addition, the alphabet for each force field can be set individually and thus potential extensions would not have to support all currently available modifications. In fact, the main effort of adding new force fields would be spent on the parameterization of the new amino acid side-chains.

CHAPTER 8

Bibliography

[1] Woltlab Burning Board Lite. `http://www.woltlab.de`. 39

[2] Ahlgren, J. A., and Ordal, G. W. Methyl esterification of glutamic acid residues of methyl-accepting chemotaxis proteins in Bacillus subtilis. *Biochem. J. 213*, 3 (Sept. 1983), 759–763. 27

[3] Alder, B. J., and Wainwright, T. E. Studies in Molecular Dynamics. I. General Method. *J. Chem. Phys. 31* (1959), 459. 46

[4] Allfrey, V. G., Faulkner, R., and Mirsky, A. E. Acetylation and methylation of histones and their possible role in the regulation of RNA synthesis. *Proc. Natl. Acad. Sci. U. S. A. 51*, 786–794. 21

[5] Arnesen, T., Van Damme, P., Polevoda, B., Helsens, K., Evjenth, R., Colaert, N., Varhaug, J., Vandekerckhove, J., Lillehaug, J., Sherman, F., et al. Proteomics analyses reveal the evolutionary conservation and divergence of N-terminal acetyltransferases from yeast and humans. *Proceedings of the National Academy of Sciences 106*, 20 (2009), 8157. 27

[6] Ashcroft, M., Kubbutat, M. H. G., and Vousden, K. H. Regulation of p53 Function and Stability by Phosphorylation. *Molecular and cellular biology 19*, 3 (1998), 1751–1758. 24

[7] Bandyopadhyay, P. Vitamin K-Dependent γ-Glutamylcarboxylation: An Ancient Posttranslational Modification. *Vitamins & Hormones 78* (2008), 157–184. 22

[8] Berendsen, H. J. C., et al. Groningen Machine for Chemical Simulations. `http://www.gromacs.org`. [accessed 15-December-2011]. 8, 92, 94

[9] Berg, R., and Prockop, D. The thermal transition of a non-hydroxylated form of collagen. Evidence for a role for hydroxyproline in stabilizing the triple-helix of collagen. *Biochemical and biophysical research communications 52*, 1 (1973), 115–120. 26

[10] Berlett, B. S., and Stadman, E. R. Protein Oxidation in Aging, Disease, and Oxidative Stress. *The Journal of Biological Chemistry 272*, 33 (1997), 20313–20316. 28, 29

[11] Beveridge, D. L., and DiCapua, F. M. Free energy via molecular simulation: applications to chemical and biomolecular systems. *Annu Rev Biophys Biophys Chem 18* (1989), 431–92. 52

[12] Birch, J. Superfish v1.4.8 – jQuery menu plugin. 36

[13] Bolton, E., et al. PubChem Substance and PubChem Compound. `http://pubchem.ncbi.nlm.nih.gov/`, 2008. 55

[14] Chemaxon. Marvinsketch version 5.5.0.1. `http://www.chemaxon.com`, 2012. [accessed 10-August-2011]. 55

[15] Clarke, S. Aging as war between chemical and biochemical processes: protein methylation and the recognition of age-damaged proteins for repair. *Ageing research reviews 2*, 3 (2003), 263–285. 27

[16] Clarke, S., and Tamanoi, F. Fighting cancer by disrupting C-terminal methylation of signaling proteins. *J. Clin. Invest. 113*, 4 (Feb. 2004), 513–515. 23

[17] Dean, R., Fu, S., Stocker, R., and Davies, M. Biochemistry and pathology of radical-mediated protein oxidation. *Biochemical Journal 324*, Pt 1 (1997), 1. 28

[18] Drew, B., and Leeuwenburgh, C. Aging and the role of reactive nitrogen species. *Annals of the New York Academy of Sciences 959*, 1 (2002), 66–81. 30

[19] Driscoll, W., Mueller, S., Eipper, B., and Mueller, G. Differential regulation of peptide α-amidation by dexamethasone and disulfiram. *Molecular pharmacology 55*, 6 (1999), 1067–1076. 23

[20] Eberhardt, E., Panasik Jr, N., and Raines, R. Inductive effects on the energetics of prolyl peptide bond isomerization: Implications for collagen folding and stability. *Journal of the American Chemical Society 118*, 49 (1996), 12261–12266. 26

[21] Eipper, B., Milgram, S., Jean Husten, E., Yun, H., and Mains, R. Peptidylglycine α-amidating monooxygenase: A multifunctional protein with catalytic, processing, and routing domains. *Protein Science 2*, 4 (1993), 489–497. 23

[22] Forte, G., Pool, M., and Stirling, C. N-terminal acetylation inhibits protein targeting to the endoplasmic reticulum. *PLoS biology 9*, 5 (2011), e1001073. 27

[23] Frenkel, D., and Smit, B. Understanding Molecular Simulation. From Algorithms to Applications. 46

[24] Fuchs, T. scriptaculous. `http://script.aculo.us`. [accessed 10-August-2011]. 36

[25] Fujiyama, A., Tsunasawa, S., Tamanoi, F., and Sakiyama, F. S-farnesylation and methyl esterification of C-terminal domain of yeast RAS2 protein prior to fatty acid acylation. *Journal of Biological Chemistry 266*, 27 (1991), 17926. 23

[26] Giglione, C., Boularot, A., and Meinnel, T. Protein N-terminal methionine excision. *Cellular and molecular life sciences 61*, 12 (2004), 1455–1474. 27

[27] Glozak, M., Sengupta, N., Zhang, X., and Seto, E. Acetylation and deacetylation of non-histone proteins. *Gene 363* (Dec. 2005), 15–23. 21

[28] Golemi, D., Maveyraud, L., Vakulenko, S., Samama, J., and Mobashery, S. Critical involvement of a carbamylated lysine in catalytic function of class D β-lactamases. *Proceedings of the National Academy of Sciences 98*, 25 (2001), 14280. 23

[29] Gorres, K. L., and Raines, R. T. Prolyl 4-hydroxylase. *Crit. Rev. Biochem. Mol. Biol. 45*, 2 (Apr. 2010), 106–124. 25

[30] Grauffel, C., Stote, R. H., and Dejaegere, A. Force field parameters for the simulation of modified histone tails. *Journal of Computational Chemistry 31*, 13 (2010), 2434–2451. 7

[31] Higgins-Gruber, S., Mutucumarana, V., Lin, P., Jorgenson, J., Stafford, D., and Straight, D. Effect of Vitamin K-dependent Protein Precursor Propeptide, Vitamin K Hydroquinone, and Glutamate Substrate Binding on the Structure and Function of γ-Glutamyl Carboxylase. *Journal of Biological Chemistry 285*, 41 (2010), 31502–31508. 22

[32] Hitakomate, E., Hood, F., Sanderson, H., and Clarke, P. The methylated N-terminal tail of RCC1 is required for stabilisation of its interaction with chromatin by Ran in live cells. *BMC cell biology 11*, 1 (2010), 43. 27

[33] Hoch, J., and Silhavy, T. *Two-component signal transduction.* Amer Society for Microbiology, 1995. 25

[34] Holmgren, S., Taylor, K., Bretscher, L., Raines, R., et al. Code for collagen's stability deciphered. *Nature 392*, 6677 (1998), 666–667. 26

[35] Hwang, C., Shemorry, A., and Varshavsky, A. N-terminal acetylation of cellular proteins creates specific degradation signals. *Science's STKE 327*, 5968 (2010), 973. 27

[36] Khorasanizadeh, S. The nucleosome: from genomic organization to genomic regulation. *Cell 116*, 2 (2004), 259–272. 26

[37] Khoury, G. A., Thompson, J. P., and Floudas, C. A. Forcefield Ptm: Development and Testing of a First Generation AMBER Forcefield for Post-Translational Modifications. 7

[38] Latham, J. A., and Dent, S. Y. R. Cross-regulation of histone modifications. *Nat. Struct. Mol. Biol. 14*, 11 (Nov. 2007), 1017–24. 22

[39] Li, J., Cross, J. B., Vreven, T., Meroueh, S. O., Mobashery, S., and Schlegel, H. B. Lysine carboxylation in proteins: OXA-10 beta-lactamase. *Proteins: Structure, Function, and Bioinformatics 61* (NOV January 2005), 246–257. 22

[40] Lippincott, J., and Apostol, I. Carbamylation of cysteine: a potential artifact in peptide mapping of hemoglobins in the presence of urea. *Analytical biochemistry 267*, 1 (1999), 57–64. 23

[41] Management Group Object. UML2.0 specification. http://www.omg.org/spec/UML/2.0/, 2012. [accessed 21-March-2012]. 34

[42] Mann, M., Ong, S. E., Grønborg, M., Steen, H., Jensen, O. N., and Pandey, A. Analysis of protein phosphorylation using mass spectrometry: deciphering the phosphoproteome. *Trends Biotechnol. 20*, 6 (June 2002), 261–268. 25

[43] Manning, G., Whyte, D., Martinez, R., Hunter, T., and Sudarsanam, S. The protein kinase complement of the human genome. *Science's STKE 298*, 5600 (2002), 1912–1934. 25

[44] Margreitter, C., Petrov, D., and Zagrovic, B. Unpublished results. 14

[45] Martinez, A., Traverso, J., Valot, B., Ferro, M., Espagne, C., Ephritikhine, G., Zivy, M., Giglione, C., and Meinnel, T. Extent of N-terminal modifications in cytosolic proteins from eukaryotes. *Proteomics 8*, 14 (2008), 2809–2831. 27

[46] Mattson, P. M. Pathways towards and away from Alzheimer's disease. *Nature 430* (2004), 631–639. 29

[47] McCoy, J., Bailey, L., Bitto, E., Bingman, C., Aceti, D., Fox, B., and Phillips Jr, G. Structure and mechanism of mouse cysteine dioxygenase. *Proceedings of the National Academy of Sciences of the United States of America 103*, 9 (2006), 3084–3089. 29

[48] Mogk, A., and Bukau, B. When the beginning marks the end. *Science's STKE 327*, 5968 (2010), 966. 27

[49] Mohiuddin, I., Chai, H., Lin, P., Lumsden, A., Yao, Q., and Chen, C. Nitrotyrosine and chlorotyrosine: clinical significance and biological functions in the vascular system. *Journal of Surgical Research 133*, 2 (2006), 143–149. 30

[50] Mydel, P., Wang, Z., Brisslert, M., Hellvard, A., Dahlberg, L., Hazen, S., and Bokarewa, M. Carbamylation-dependent activation of T cells: A novel mechanism in the pathogenesis of autoimmune arthritis. *The Journal of Immunology 184*, 12 (2010), 6882–6890. 23

[51] Negishi, M., Pedersen, L., Petrotchenko, E., Shevtsov, S., Gorokhov, A., Kakuta, Y., and Pedersen, L. Structure and function of sulfotransferases. *Archives of biochemistry and biophysics 390*, 2 (2001), 149–157. 30

[52] NIST. Expect: A tool for automating interactive applications. `http://www.nist.gov/el/msid/expect.cfm`. [accessed 19-June-2012]. 105

[53] Nokelainen, M., Helaakoski, T., Myllyharju, J., Notbohm, H., Pihlajaniemi, T., Fietzek, P., and Kivirikko, K. Expression and characterization of recombinant human type II collagens with low and high contents of hydroxylysine and its glycosylated forms. *Matrix biology 16*, 6 (1998), 329–338. 25

[54] Oostenbrink, C., Juchli, D., and van Gunsteren, W. F. Amine Hydration: A United-Atom Force-Field Solution. *ChemPhysChem 6* (2005), 1800–1804. 56

[55] Oostenbrink, C., Soares, T. A., van der Vegt, N. F. A., and van Gunsteren, W. F. Validation of the 53A6 GROMOS force field. *Eur Biophys J 34* (2005), 273–284. 51

[56] Oostenbrink, C., Villa, A., Mark, A. E., and van Gunsteren, W. F. A Biomolecular Force Field Based on the Free Enthalpy of Hydration and Solvation: The GROMOS Force-Field Parameter Sets 53A5 and 53A6. *J Comput Chem 25* (2004), 1656–1676. 51, 52

[57] Oracle. MySQL: open source database. `http://www.mysql.com`. [accessed 19-June-2012]. 111

[58] Ozdemir, A., et al. Characterization of Lysine 56 of Histone H3 as an Acetylation Site in Saccharomyces cerevisiae. *The Journal of Biological Chemistry 280* (2005), 25949–25952. 21

[59] Paik, W., Paik, D., and Kim, S. Historical review: the field of protein methylation. *Trends in biochemical sciences 32*, 3 (2007), 146–152. 26

[60] Petrov, D., Margreitter, C., Oostenbrink, C., Grandits, M., and Zagrovic, B. Unpublished results. 55

[61] Petrov, D., and Zagrovic, B. Microscopic analysis of protein oxidative damage: effect of carbonylation on structure, dynamics, and aggregability of villin headpiece. *J. Am. Chem. Soc. 133*, 18 (May 2011), 7016–24. 53

[62] RCSB, et al. wwPDB - Worldwide Protein Data Bank. `http://www.wwpdb.org`. [accessed 15-December-2011]. 90

[63] Research Collaboratory for Structural Bioinformatics - RCSB. PDB - Protein Data Bank. http://www.pdb.org. [accessed 15-December-2011]. 8

[64] Rodrigue, A., Quentin, Y., Lazdunski, A., Méjean, V., and Foglino, M. Two-component systems in Pseudomonas aeruginosa: why so many? *Trends in microbiology 8*, 11 (2000), 498–504. 24

[65] Rogaeva, E., Meng, Y., Lee, J. H., Gu, Y., Kawarai, T., Zou, F., Katayama, T., Baldwin, C. T., Cheng, R., Hasegawa, H., Chen, F., Shibata, N., Lunetta, K. L., Pardossi-Piquard, R., Bohm, C., Wakutani, Y., Cupples, L. A., Cuenco, K. T., Green, R. C., Pinessi, L., Rainero, I., Sorbi, S., Bruni, A., Duara, R., Friedland, R. P., Inzelberg, R., Hampe, W., Bujo, H., Song, Y.-Q., Andersen, O. M., Willnow, T. E., Graff-Radford, N., Petersen, R. C., Dickson, D., Der, S. D., Fraser, P. E., Schmitt-Ulms, G., Younkin, S., Mayeux, R., Farrer, L. A., and George-Hyslop, P. S. The neuronal sortilin-related receptor SORL1 is genetically associated with Alzheimer disease. *Nat. Genet. 39*, 2 (Feb. 2007), 168–77. 29

[66] Royal Society of Chemistry. ChemSpider. http://www.chemspider.com. [accessed 22-March-2012]. 55

[67] Schmid, N., Eichenberger, A. P., Choutko, A., Riniker, S., Winger, M., Mark, A. E., and van Gunsteren, W. F. Definition and testing of the GROMOS force-field versions 54A7 and 54B7. *Eur Biophys J 40* (2011), 843–856. 51

[68] Schuler, L. D., Daura, X., and van Gunsteren, W. F. An Improved GROMOS96 Force Field for Aliphatic Hydrocarbons in the Condensed Phase. *Journal of Computational Chemistry 22*, 11 (2001), 1205–1218. 51

[69] Sedgwick, B., Bates, P. A., Paik, J., Jacobs, S. C., and Lindahl, T. Repair of alkylated DNA: recent advances. *DNA Repair (Amst.) 6*, 4 (Apr. 2007), 429–42. 27

[70] Seger, R., and Krebs, E. G. The MAPK signaling cascade. *FASEB Journal 9*, 726–735. 24

[71] Seibert, C., and Sakmar, T. Toward a framework for sulfoproteomics: Synthesis and characterization of sulfotyrosine-containing peptides. *Peptide Science 90*, 3 (2008), 459–477. 30

[72] Smith, W., Gilboe, D., and Henderson, L. Incorporation of Hydroxylysine into the Cell Wall and a Cell-Wall Precursor in Staphylococcus aureus. *Journal of bacteriology 89*, 1 (1965), 136–140. 25

[73] Smith, W., Schurter, B., Wong-Staal, F., and David, M. Arginine methylation of RNA helicase a determines its subcellular localization. *Journal of Biological Chemistry 279*, 22 (2004), 22795–22798. 27

[74] Soares, T. A., Daura, X., Oostenbrink, C., Smith, L. J., and van Gunsteren, W. F. Validation of the GROMOS force-field parameter set 45a3 against nuclear magnetic resonance data of the hen egg lysozyme. *Journal of Biomolecular NMR 30* (2004), 407–422. 51

[75] Stock, A., Clarke, S., Clarke, C., and Stock, J. N-terminal methylation of proteins: structure, function and specificity. *FEBS letters 220*, 1 (1987), 8–14. 27

[76] Suttie, J. W. Vitamin K-dependent carboxylase. *Annu. Rev. Biochem. 54* (1985), 459–77. 22

[77] Turner, B. Cellular memory and the histone code. *Cell 111*, 3 (2002), 285–291. 26

[78] Van Der Spoel, D., Lindahl, E., Hess, B., Van Buuren, A., Apol, E., Meulenhoff, P., Tieleman, D., Sijbers, A., Feenstra, K., Van Drunen, R., Berendsen, H. J. C., et al. Gromacs User Manual version 4.5.4. 46, 47

[79] van Gunsteren, W. F. Biomolecular Modeling: Goals, Problems, Perspectives. *Review in Angew. Chem. Int.* (2006), 4064–4092. 46

[80] Van Slyke, D., and Hiller, A. An unidentified base among the hydrolytic products of gelatin. *Proceedings of the National Academy of Sciences of the United States of America 7*, 7 (1921), 185. 25

[81] Walsh, C., Garneau-Tsodikova, S., and Gatto Jr, G. Protein posttranslational modifications: the chemistry of proteome diversifications. *Angewandte Chemie International Edition 44*, 45 (2005), 7342–7372. 7, 20, 21, 22, 24, 25, 26, 27

[82] Walsh, C. T. Posttranslational modification of proteins: Expanding nature's inventory. 7

[83] Wold, F. In vivo chemical modification of proteins (post-translational modification). *Annual review of biochemistry 50*, 1 (1981), 783–814. 28

[84] Wurtele, H., et al. Histone H3 Lysine 56 Acetylation and the Response to DNA Replication Fork Damage. *Mol. Cell. Biol. 32*, 11 (Jan. 2012), 154–172. 21

[85] Yang, X., and Seto, E. Lysine acetylation: codified crosstalk with other posttranslational modifications. *Molecular cell 31*, 4 (2008), 449–461. 21, 22

[86] Zhang, F., and Casey, P. Protein prenylation: molecular mechanisms and functional consequences. *Annual review of biochemistry 65*, 1 (1996), 241–269. 23

CHAPTER 9

Appendix

9.1 Maintenance instructions

9.1.1 Add modification

In order to add a modification to an implemented force field, it is necessary to alter several files parsed by the server. Each modification is described by a triplet of parameters: the residue it is applied to, the internal four-letter code and the force field version considered[1].

9.1.2 Add existing modification type to another amino acid

If a type of modification has been already implemented in the workflow and has to be added to another amino acid, the procedure is as follows:

1. Add the new building block to the correct topology file to provide the required parameters[2].

2. Get the correct modification ID from the list[9.2] and add an entry in the parameters file (for details see subsection 6.1.2) to provide the backend with the required instructions. This includes e.g. the name of the target amino acid, the new name after alteration and specifications for deleted or added atoms. It is absolutely crucial to add the residue block to the proper modification type block. Access is possible via the admin panel[4.1.1.7] after successful authentication.

3. To allow selection of this new modification, the modification abbreviation has to be added to the database holding the information, which residue is capable of which set of modifications. Use the admin panel to add the four-letter code of the desired modification to the list of modifications for the particular force field and residue.

9.1.3 Add new modification type

In order to add a completely new type of modifications to an implemented force field, it is necessary to alter several files parsed by the server. Prior to the steps described above, the new modification block has to be added in the force field's configuration file (see description in subsection 6.1.2). The identifier of the modification should be of size three letters (both numbers and letters are allowed) to be consistent with the PDB file format. This identifier has to be added to the list of the building blocks (see 6.3.5) and the number in the first line has to be updated, too. Furthermore, the modification type has to be set in the script `display_residues_form.php` in order enable it in the menu.

[1] As for example, the pair `HYDC` and aspartic acid will lead to a charged hydroxylated aspartic acid, while the same residue with `HYCS` will result in its stereoversion. Therefore, `HYDC` is the block in the configuration file[6.1.2] and within that block, there is an entry for each relevant amino acid. The force field is specified by the database filename.

[2] In case of GROMACS, these are the rtp files[6.3.4].

9.2 Supplementary material

9.2.1 Terminal menus in pdb2gmx

Since it is not possible to specify terminal modifications automatically in the initial call of the executable, `pdb2gmx` has to be fed with these arguments. Therefore, the backend transiently generates an `expect` instruction file [52] in order to simulate manual selection by the user. Note, that this functionality might be dependent on the GROMACS version installed and thus it should be checked in case of an update. Moreover, it is extremely vulnerable to changes in the N- and C-terminal files since the entries are ordered alphabetically and not by occurrence in the file. It is crucial to ensure that the related entries in the modification instruction file are consistent.

N-terminal menus for pdb2gmx			
proline	**glycine**	**methionine**	**rest**
0: PRO-NH2+	0: GLY-NH3+	0: MET-FOR	0: NH3+
1: PRO-NH	1: GLY-1NM	1: NH3+	1: NH2
2: PRO-NAC	2: GLY-1NC	2: NH2	2: NAC
3: PRO-1NM	3: GLY-2NM	3: NAC	3: PGA
4: PRO-1NC	4: GLY-2NC	4: PGA	4: 1NM
5: PRO-2NC	5: GLY-3NC	5: 1NM	5: 1NC
6: NH3+	6: NH2	6: 1NC	6: 2NM
7: PGA	7: NAC	7: 2NM	7: 2NC
8: 2NM	8: PGA	8: 2NC	8: 3NC
9: 3NC	9: None	9: 3NC	9: None
10: None		10: None	
C-terminal menu for pdb2gmx			
All			
0: COO-			
1: COOH			
2: AMD			
3: CME			
4: None			

Table 9.1: Representation of the menus for terminal modifications performed by `pdb2gmx`.

9.2.2 List of modifications and corresponding groups

Complete list of modifications		
Name	**Class**	**Description**
		acetylation
[KAC]	ACET	acetyllysine
		carboxylation / carbamylation
[ECN]	CARN	carboxyglutamic acid (-1)
[ECA]	CARB	carboxyglutamic acid (-2)
[KCN]	CARN	carboxylysine (0)
[KCA]	CARB	carboxylysine (-1)
[CAM]	CAAM	carbamylated cysteine
[KAM]	CAAM	carbamylated lysine
		hydroxylation (enzymatic)
[P3H]	HY3S	3'-hydroxyproline (S)
[PH3]	HYD3	3'-hydroxyproline (R)
[HYP]	HY4S	4'-hydroxyproline (S)
[HY2]	HYD4	4'-hydroxyproline (R)
[PHH]	HYDD	3',4'-dihydroxyproline
[K6H]	HYDS	hydroxylysine (0, S)
[KH6]	HYDR	hydroxylysine (0, R)
[KHP]	HYCS	hydroxylysine (+1, S)
[KPH]	HYDC	hydroxylysine (+1, R)
[HTY]	HYDR	3',4'-dihydroxyphenylalanine
[W5H]	HYD5	5'-hydroxytryptophan
[D3H]	HYCS	3'-hydroxyaspartic acid (-1, S)
[DH3]	HYDC	3'-hydroxyaspartic acid (-1, R)
[D3N]	HYDS	3'-hydroxyaspartic acid (0, S)

`[DN3]`	HYDR	3'-hydroxyaspartic acid (0, R)
`[N3H]`	HYDR	3'-hydroxyasparagine (S)
`[NH3]`	HYDS	3'-hydroxyasparagine (R)
`[TYR]`	HYD4	4'-hydroxyphenylalanine (tyrosine)
hydroxylation (non-enzymatic)		
`[F23]`	HYDD	2',3'-dihydroxyphenylalanine
`[F2H]`	HYD2	2'-hydroxyphenylalanine
`[F3H]`	HYD3	3'-hydroxyphenylalanine
`[W7H]`	HYD7	7'-hydroxytryptophan
`[W6H]`	HYD6	6'-hydroxytryptophan
`[W4H]`	HYD4	4'-hydroxytryptophan
`[W2H]`	HYD2	2'-hydroxytryptophan
`[L3H]`	HYD3	3'-hydroxyleucine (R)
`[LH3]`	HY3S	3'-hydroxyleucine (S)
`[L4H]`	HYD4	4'-hydroxyleucine
`[L5H]`	HYD5	5'-hydroxyleucine (R)
`[LH5]`	HY5S	5'-hydroxyleucine (S)
`[V3H]`	HYD3	3'-hydroxyvaline
`[CYH]`	HYDR	hydroxycysteine
`[P5H]`	HY5S	5'-hydroxyproline (S)
`[PH5]`	HYD5	5'-hydroxyproline (R)
methylation		
`[KMN]`	METY	methyllysine (0)
`[KMC]`	ME1C	methyllysine (+1)
`[K2M]`	MET2	dimethyllysine (0)
`[K2C]`	ME2C	dimethyllysine (+1)
`[K3C]`	MET3	trimethyllysine

[RMN]	METY	methylarginine (0)
[RMC]	ME1C	methylarginine (+1)
[RSM]	MS2N	dimethylarginine (0, sym)
[RMS]	MS2C	dimethylarginine (+1, sym)
[RAM]	MA2N	dimethylarginine (0, asy)
[RMA]	MA2C	dimethylarginine (+1, asy)
[H1M]	M11N	1'-methylhistidine (0)
[H1C]	M11C	1'-methylhistidine (+1)
[H3M]	M13N	3'-methylhistidine (0)
[H3C]	M13C	3'-methylhistidine (+1)
[QME]	METY	methylglutamine
[NME]	METY	methylasparagine
[EME]	METY	methylglutamic acid
[DME]	METY	methylaspartic acid
[CYM]	METY	methylcysteine
		miscellaneous
[YSU]	SULF	sulfotyrosine
[LNO]	NORL	norleucine (structural isomer of leucine)
[GSA]	GSAL	glutamic semialdehyde
[KAL]	ALLY	allysine
[RCI]	CITR	citrulline
[SDH]	DEHY	dehydroserine
[TDH]	DEHY	dehydrothreonine
[WBR]	BROM	bromotryptophan
[WKY]	KYNU	kynurenine
[WKH]	HYKY	3'-hydroxykynurenine
[WKF]	FOKY	formylkynurenine

[NNG]	NGLY	N-glycosylated asparagine
[YCH]	CHLO	chlorotyrosine
[CSN]	NITR	S-nitrocysteine
[YNI]	NITR	nitrotyrosine
[WNI]	NITR	6’-nitrotryptophan
[CSE]	SULF	sulfocysteine
[MES]	SULF	sulfomethionine
oxidation		
[PGA]	OXID	pyroglutamic acid
[H2X]	OXID	2’-oxohistidine
[TOX]	OXID	oxothreonine
[MSX]	OXID	sulfoxide methionine
[MXS]	OXIS	sulfoxide methionine (sv)
[CSA]	OXID	cysteine sulfinic acid
phosphorylation		
[S1P]	PHOS	phosphoserine (-1)
[S2P]	PHO2	phosphoserine (-2)
[T1P]	PHOS	phosphothreonine (-1)
[T2P]	PHO2	phosphothreonine (-2)
[Y1P]	PHOS	phosphotyrosine (-1)
[Y2P]	PHO2	phosphotyrosine (-2)
[D1P]	PHOS	phosphoaspartic acid (-1)
[D2P]	PHO2	phosphoaspartic acid (-2)
[K1P]	PHOS	6’-phospholysine (-1)
[K2P]	PHO2	6’-phospholysine (-2)
[R1P]	PHOS	phosphoarginine (-1)
[R2P]	PHO2	phosphoarginine (-2)

[H11]	PH11	1'-phosphohistidine (-1)
[H12]	PH12	1'-phosphohistidine (-2)
[H31]	PH31	3'-phosphohistidine (-1)
[H32]	PH32	3'-phosphohistidine (-2)
N-terminal modification		
[NAC]	NACE	acetylation: all amino acids
[PGA]	NPYC	pyroglutamic acid: glutamic acid, glutamine
[MFO]	NFOM	formylation: methionine
[PYA]	NPYA	pyruvic acid: serine, cysteine, valine
[1NM]	NMEN	methylation (0): all amino acids
[1NC]	NMEC	methylation (+1): all amino acids
[2NM]	NM2N	dimethylation (0): all amino acids
[2NC]	NM2C	dimethylation (+1): all amino acids
[3NC]	NM3C	trimethylation: all amino acids
C-terminal modification		
[CME]	CMET	methylation: cysteine, lysine, leucine
[AMD]	CAMD	amidation: all amino acids

Table 9.2: All modifications presented in the present work together with their corresponding internal modification class. In principle, there is no restriction for the names of these classes, whereas those of the residues should be of length three. For details regarding abbreviations, see subsection 2.1.2.

9.3 Database tables

The database used by the frontend for job information handling, statistics *et cetera* is a `MySQL` database [57]. The following tables are part of database `re_webserver_DB`. Text fields are encoded in *latin1_swedish_ci*. Note, that each day a complete database backup is made which can be used to restore the server in case of data corruption or reinstallation.

9.3.1 AUTH_LOGIN

Table `AUTH_LOGIN` contains all the information regarding users (including administrators). Passwords are hashed before they are stored.

AUTH_LOGIN				
Field	Type [length]	Default	Extra	Primary
user_ID	int [11]		auto_increment	yes
user_Name	varchar [25]			
user_PWD	varchar [400]			
user_Group	varchar [10]	USER		
user_TS	timestamp	CURRENT_TIMESTAMP		
user_Email	varchar [250]			

Table 9.3: User entries. They are read by the frontend scripts when logging in. New users can be added via the admin panel.

9.3.2 POSSIBLE_MODS

For each combination of canonical residues and force fields, table `POSSIBLE_MODS` holds an entry specifying, which modifications are available.

POSSIBLE_MODS				
Field	Type [length]	Default	Extra	Primary
entry_ID	int [11]		auto_increment	yes
ForceField	varchar [30]			
Residue	varchar [4]			
SerializedArray	text			

Table 9.4: Available modifications. Table stores four-letter IDs of modification classes in serialized arrays. They are parsed by `display_residues_form.php` and can be edited by the `dbs` script in the admin panel.

9.3.3 AUTH_USER_PERMISSIONS

All the data relevant for a particular job, is stored in table AUTH_USER_PERMISSIONS and parsed at each script call by the header.php component.

AUTH_USER_PERMISSIONS				
Field	Type [length]	Default	Extra	Primary
IND	int [11]		auto_increment	yes
KEYSTR	varchar [200]			
E_MAIL	varchar [250]			
FILESTR	varchar [500]			
ACTDATE	date			
SUCCESS	tinyint [1]			
ACTTIME	time			
IPADDRESS	varchar [15]			
PATHSTR	varchar [200]			
FFTYPE	varchar [50]			
COPYHEADER	varchar [9]			
DEBUMP	varchar [9]			
MINIMIZE	varchar [9]			
IFTYPE	varchar [9]			
PROC_SCRIPT	varchar [1000]			
RESTRAINTS	varchar [9]			
RES_STRENGTH	varchar [9]			

Table 9.5: Job data. In case job files are deleted (either by the user or by general maintenance), the database entries are retained for statistics.

9.3.4 SITE_DOWNLOADS

All provided downloadlinks are encoded in this table.

SITE_DOWNLOADS				
Field	Type [length]	Default	Extra	Primary
download_ID	int [11]		auto_increment	yes
download_Path	varchar [500]			
download_Title	varchar [200]			
download_Description	text			
download_Type	varchar [100]			
download_Version	varchar [40]			
download_Counter	int [11]	0		

Table 9.6: Downloadlinks. This table provides a layer hiding the actual storage site on the server. For authenticated administrators, editing and deletion of individual downloads can be facilitated directly by the responsible script (`downloads.php`).

9.3.5 SITE_STUFF

In this table, miscellaneous settings are stored (see below). It allows central buffering of frontend-wide parameters such as the general switch for setting minimization or refreshing intervals.

SITE_STUFF				
Field	Type [length]	Default	Extra	Primary
stuff_ID	int [11]		auto_increment	yes
stuff_Name	varchar [200]			
stuff_Value	text			
stuff_lastChange	date			
stuff_Flag	tinyint [1]			

Table 9.7: Miscellaneous settings. Depending on the particular case, the flag is considered or not. Note that all changes are mediated by usage of class `stuff` objects.

9.4 Set of available parameters for ffG45a3

The following lists are directly taken from the corresponding files in GROMACS for the sake of completeness. See subsection 4.2.2 for a more detailed description of both parameters and potentials.

9.4.1 Atomtypes

Listing 9.1: Atomtypes

```
     O  15.99940 ; carbonyl oxygen (C=O)
    OM  15.99940 ; carboxyl oxygen (CO-)
    OA  15.99940 ; hydroxyl, sugar or ester oxygen
    OW  15.99940 ; water oxygen
     N  14.00670 ; peptide nitrogen (N or NH)
    NT  14.00670 ; terminal nitrogen (NH2)
    NL  14.00670 ; terminal nitrogen (NH3)
    NR  14.00670 ; aromatic nitrogen
    NZ  14.00670 ; Arg NH (NH2)
    NE  14.00670 ; Arg NE (NH)
     C  12.01100 ; bare carbon
   CH0  12.01100 ; tetrahedral aliphatic bare carbon ref.  1
   CH1  13.01900 ; aliphatic or sugar CH-group
   CH2  14.02700 ; aliphatic or sugar CH2-group
   CH3  15.03500 ; aliphatic CH3-group
   CH4  16.04300 ; methane
  CH2r  14.02700 ; cycloalkanic CH2-group ref.  1
   CR1  13.01900 ; aromatic CH-group
    HC   1.00800 ; hydrogen bound to carbon
     H   1.00800 ; hydrogen not bound to carbon
   DUM   0.00000 ; dummy atom, no idea what the mass should be.  PT3-99
     S  32.06000 ; sulfur
  CU1+  63.54600 ; copper (charge 1+)
  CU2+  63.54600 ; copper (charge 2+)
    FE  55.84700 ; iron (heme)
  ZN2+  65.37000 ; zinc (charge 2+)
  MG2+  24.30500 ; magnesium (charge 2+)
  CA2+  40.08000 ; calcium (charge 2+)
     P  30.97380 ; phosphor
    AR  39.94800 ; argon
     F  18.99840 ; fluor (non-ionic)
    CL  35.45300 ; chlorine (non-ionic)
    BR  79.90400 ; bromine (non-ionic)
  CMet  15.035   ; CH3-group in methanol (solvent)
  OMet  15.9994  ; oxygen in methanol (solvent)
   NA+  22.9898  ; sodium (charge 1+)
   CL-  35.45300 ; chlorine (charge 1-)
  CChl  12.011   ; carbon in chloroform (solvent)
 CLChl  35.453   ; chloride in chloroform (solvent)
  HChl  1.008    ; hydrogen in chloroform (solvent)
 SDmso  32.06000 ; DMSO Sulphur (solvent)
 CDmso  15.03500 ; DMSO Carbon (solvent)
 ODmso  15.99940 ; DMSO Oxygen (solvent)
  CCl4  12.011   ; carbon in carbontetrachloride (solvent)
 CLCl4  35.453   ; chloride in carbontetrachloride (solvent)
    SI  28.08    ; silicon
  MNH3   0       ; Dummy mass in rigid tetraedrical NH3 group
    MW   0       ; Dummy mass in rigid tyrosine rings
```

9.4.2 Bonded interactions

Listing 9.2: Bond-stretching parameters

```
; GROMOS bond-stretching parameters
;
; Bond type code
; Force constant
; Ideal bond length
; Examples of usage in terms of non-bonded atom types
;
; ICB(H)[N] CB[N] B0[N]
;
#define gb_1        0.1000  1.5700e+07
; H - OA 750
#define gb_2        0.1000  1.8700e+07
; H - N (all) 895
#define gb_3        0.1090  1.2300e+07
; HC - C 700
#define gb_4        0.1230  1.6600e+07
; C - O 1200
#define gb_5        0.1250  1.3400e+07
; C - OM 1000
#define gb_6        0.1320  1.2000e+07
; CR1 - NR (6-ring) 1000
#define gb_7        0.1330  8.8700e+06
; H - S 750
#define gb_8        0.1330  1.0600e+07
; C - NT, NL 900
#define gb_9        0.1330  1.1800e+07
; C, CR1 - N, NR, CR1, C (peptide, 5-ring) 1000
#define gb_10       0.1340  1.0500e+07
; C - N, NZ, NE 900
#define gb_11       0.1340  1.1700e+07
; C - NR (no H) (6-ring) 1000
#define gb_12       0.1360  1.0200e+07
; C - OA 900
#define gb_13       0.1380  1.1000e+07
; C - NR (heme) 1000
#define gb_14       0.1390  8.6600e+06
; CH2 - C, CR1 (6-ring) 800
#define gb_15       0.1390  1.0800e+07
; C, CR1 - CH2, C, CR1 (6-ring) 1000
#define gb_16       0.1400  8.5400e+06
; C, CR1, CH2 - NR (6-ring) 800
#define gb_17       0.1430  8.1800e+06
; CHn - OA 800
#define gb_18       0.1430  9.2100e+06
; CHn - OM 900
#define gb_19       0.1435  6.1000e+06
; CHn - OA (sugar) 600
#define gb_20       0.1470  8.7100e+06
; CHn - N, NT, NL, NZ, NE 900
#define gb_21       0.1480  5.7300e+06
; CHn - NR (5-ring) 600
#define gb_22       0.1480  7.6400e+06
; CHn - NR (6-ring) 800
#define gb_23       0.1480  8.6000e+06
; O, OM - P 900
#define gb_24       0.1500  8.3700e+06
```

```
; O - S 900
#define gb_25       0.1520  5.4300e+06
; CHn - CHn (sugar) 600
#define gb_26       0.1530  7.1500e+06
; C, CHn - C, CHn 800
#define gb_27       0.1610  4.8400e+06
; OA - P 600
#define gb_28       0.1630  4.7200e+06
; OA - SI 600
#define gb_29       0.1780  5.9400e+06
; CH3 - S 900
#define gb_30       0.1830  5.6200e+06
; CH2 - S 900
#define gb_31       0.1870  3.5900e+06
; CH1 - SI 600
#define gb_32       0.1980  0.6400e+06
; NR - FE 120
#define gb_33       0.2040  5.0300e+06
; S - S 1000
#define gb_34       0.2000  0.6280e+06
; NR (heme) - FE 120
#define gb_35       0.1000  2.3200e+07
; HWat - OWat 1110
#define gb_36       0.1100  1.2100e+07
; HChl - CChl 700
#define gb_37       0.1758  8.1200e+06
; CChl - CLChl 1200
#define gb_38       0.1530  8.0400e+06
; ODmso - SDmso 900
#define gb_39       0.1950  4.9500e+06
; SDmso - CDmso 900
#define gb_40       0.1760  8.1000e+06
; CCl4 - CLCl4 1200
#define gb_41     0.163299  8.7100e+06
; HWat - HWat 1110
#define gb_42     0.233839  2.6800e+06
; HChl - CLChl 700
#define gb_43     0.290283  2.9800e+06
; CLChl - CLChl 1200
#define gb_44     0.280412  2.3900e+06
; ODmso - CDmso 900
#define gb_45     0.292993  2.1900e+06
; CDmso - CDmso 900
#define gb_46     0.198842  3.9700e+06
; HMet - CMet 750
#define gb_47     0.287407  3.0400e+06
; CLCl4 - CLCl4 1200
```

Listing 9.3: Bond-angle bending parameters

```
; GROMOS bond-angle bending parameters
;
;
; Bond-angle type code
; Force constant
; Ideal bond angle
; Example of usage in terms of non-bonded atom types
;
;
; ICT(H)[N] CT[N] (T0[N])
;
#define ga_1         90.00      420.00
; NR(heme) - FE - NR(heme) 100
#define ga_2         96.00      405.00
; H - S - CH2 95
#define ga_3        100.00      475.00
; CH2 - S - CH3 110
#define ga_4        103.00      420.00
; OA - P - OA 95
#define ga_5        104.00      490.00
; CH2 - S - S 110
#define ga_6        108.00      465.00
; NR, C, CR1(5-ring) 100
#define ga_7        109.50      285.00
; CHn - CHn - CHn, NR(6-ring) (sugar) 60
#define ga_8        109.50      320.00
; CHn, OA - CHn - OA, NR(ring) (sugar) 68
#define ga_9        109.50      380.00
; H - NL, NT - H, CHn - OA - CHn(sugar) 80
#define ga_10       109.50      425.00
; H - NL - C, CHn H - NT - CHn 90
#define ga_11       109.50      450.00
; X - OA, SI - X 95
#define ga_12       109.50      520.00
; CHn,C - CHn - C, CHn, OA, OM, N, NE 110
#define ga_13       109.60      450.00
; OM - P - OA 95
#define ga_14       111.00      530.00
; CHn - CHn - C, CHn, OA, NR, NT, NL 110
#define ga_15       113.00      545.00
; CHn - CH2 - S 110
#define ga_16       115.00       50.00
; NR(heme) - FE - NR 10
#define ga_17       115.00      460.00
; H - N - CHn 90
#define ga_18       115.00      610.00
; CHn, C - C - OA, N, NT, NL 120
#define ga_19       116.00      465.00
; H - NE - CH2 90
#define ga_20       116.00      620.00
; CH2 - N - CH1 120
#define ga_21       117.00      635.00
; CH3 - N - C, CHn - C - OM 120
#define ga_22       120.00      390.00
; H - NT, NZ, NE - C 70
#define ga_23       120.00      445.00
; H - NT, NZ - H 80
#define ga_24       120.00      505.00
; H - N - CH3, H, HC - 6-ring, H - NT - CHn 90
#define ga_25       120.00      530.00
```

```
; P, SI - OA - CHn, P 95
#define ga_26       120.00      560.00
; N, C, CR1 (6-ring, no H) 100
#define ga_27       120.00      670.00
; NZ - C - NZ, NE 120
#define ga_28       120.00      780.00
; OM - P - OM 140
#define ga_29       121.00      685.00      120.00      560.00
; N, C, CR1 (6-ring, no H) 100
#define ga_27       120.00      670.00
; NZ - C - NZ, NE 120
#define ga_28       120.00      780.00
; OM - P - OM 140
#define ga_29       121.00      685.00
; O - C - CHn, C CH3 - N - CHn 120
#define ga_30       122.00      700.00
; CH1, CH2 - N - C 120
#define ga_31       123.00      415.00
; H - N - C 70
#define ga_32       124.00      730.00
; O - C - OA, N, NT, NL C - NE - CH2 120
#define ga_33       125.00      375.00
; FE - NR - CR1 (5-ring) 60
#define ga_34       125.00      750.00
; - 120
#define ga_35       126.00      575.00
; H, HC - 5-ring 90
#define ga_36       126.00      640.00
; X(noH) - 5-ring 100
#define ga_37       126.00      770.00
; OM - C - OM 120
#define ga_38       132.00      760.00
; 5, 6 ring connection 100
#define ga_39       155.00     2215.00
; SI - OA - SI 95
#define ga_40       109.50      434.00
; HWat - OWat - HWat 92
#define ga_41       107.57      484.00
; HChl - CChl - CLChl 105
#define ga_42       111.30      632.00
; CLChl - CChl - CLChl 131
#define ga_43        97.40      469.00
; CDmso - SDmso - CDmso 110
#define ga_44       106.75      503.00
; CDmso - SDmso - ODmso 110
#define ga_45       108.53      443.00
; HMet - OMet - CMet 95
#define ga_46       109.50      618.00
; CLCl4 - CCl4 - CLCl4 131
; O - C - CHn, C CH3 - N - CHn 120
#define ga_30       122.00      700.00
; CH1, CH2 - N - C 120
#define ga_31       123.00      415.00
; H - N - C 70
#define ga_32       124.00      730.00
; O - C - OA, N, NT, NL C - NE - CH2 120
#define ga_33       125.00      375.00
; FE - NR - CR1 (5-ring) 60
#define ga_34       125.00      750.00
; - 120
#define ga_35       126.00      575.00
```

```
; H, HC - 5-ring 90
#define ga_36        126.00      640.00
; X(noH) - 5-ring 100
#define ga_37        126.00      770.00
; OM - C - OM 120
#define ga_38        132.00      760.00
; 5, 6 ring connection 100
#define ga_39        155.00     2215.00
; SI - OA - SI 95
#define ga_40        109.50      434.00
; HWat - OWat - HWat 92
#define ga_41        107.57      484.00
; HChl - CChl - CLChl 105
#define ga_42        111.30      632.00
; CLChl - CChl - CLChl 131
#define ga_43         97.40      469.00
; CDmso - SDmso - CDmso 110
#define ga_44        106.75      503.00
; CDmso - SDmso - ODmso 110
#define ga_45        108.53      443.00
; HMet - OMet - CMet 95
#define ga_46        109.50      618.00
; CLCl4 - CCl4 - CLCl4 131
```

Listing 9.4: Improper dihedral parameters

```
; GROMOS improper (harmonic) dihedral angle parameters
;
;
; Improper dihedral-angle type code
; Force constant
; Ideal improper dihedral angle
; Example of usage
;
;
; ICQ(H)[N] CQ[N] (Q0[N])
;
#define gi_1            0.0   167.42309
; planar groups 40
;
#define gi_2       35.26439   334.84617
; tetrahedral centres 80
;
#define gi_3            0.0   669.69235
; heme iron 160
```

Listing 9.5: Dihedral parameters

```
; GROMOS (trigonometric) dihedral torsional angle parameters
;
;
; Dihedral-angle type code
; Force constant
; Phase shift
; Multiplicity
; Example of usage in terms of non-bonded atom types
;
;
; ICP(H)[N] CP[N] PD[N] NP[N]
;
#define gd_1    180.000       5.86          2
; -C-C- 1.4
#define gd_2    180.000       7.11          2
; -C-OA- (at ring) 1.7
#define gd_3    180.000       16.7          2
; -C-OA- (carboxyl) 4.0
#define gd_4    180.000       33.5          2
; -C-N, NT, NE, NZ,NR- 8.0
#define gd_5    180.000       41.8          2
; -C-CR1- (6-ring) 10.0
#define gd_6      0.000        0.0          2
; -CH1 (sugar)-NR(base)- 0.0
#define gd_7      0.000      0.418          2
; O-CH1-CHn-no O 0.1
#define gd_8      0.000       2.09          2
; O-CH1-CHn-O 0.5
#define gd_9      0.000       3.14          2
; -OA-P- 0.75
#define gd_10     0.000       16.7          2
; -S-S- 4.0
#define gd_11     0.000       1.05          3
; -OA-P- 0.25
#define gd_12     0.000       1.26          3
; -CHn-OA(no sugar)- 0.3
#define gd_13     0.000       2.93          3
; -CH2-S- 0.7
#define gd_14     0.000       3.77          3
; -C,CHn,SI-NT,NL,OA(sugar)- 0.9
#define gd_15     0.000       4.18          3
; HC-C-S- 1.0
#define gd_16     0.000       5.44          3
; HC-C-C- 1.3
#define gd_17     0.000       5.92          3
; -CHn,SI-CHn- 1.4
#define gd_18     0.000        0.0          4
; -NR-FE- 0.0
#define gd_19   180.000        1.0          6
; -CHn-N,NE- 0.24
#define gd_20     0.000        1.0          6
; -CHn-C,NR (ring), CR1- 0.24
#define gd_21     0.000       3.77          6
; -CHn-NT- 0.9
```

9.4.3 Non-bonded interactions

Listing 9.6: Dihedral parameters

```
[ atomtypes ]
; name at.num mass charge ptype c6 c12
    O    8      0.000      0.000      A 0.0022619536  7.4149321e-07
   OM    8      0.000      0.000      A 0.0022619536  7.4149321e-07
   OA    8      0.000      0.000      A 0.0022619536  1.505529e-06
   OW    8      0.000      0.000      A 0.0026173456  2.634129e-06
    N    7      0.000      0.000      A 0.0024364096  1.692601e-06
   NT    7      0.000      0.000      A 0.0024364096  1.692601e-06
   NL    7      0.000      0.000      A 0.0024364096  1.692601e-06
   NR    7      0.000      0.000      A 0.0024364096  3.389281e-06
   NZ    7      0.000      0.000      A 0.0024364096  1.692601e-06
   NE    7      0.000      0.000      A 0.0024364096  1.692601e-06
    C    6      0.000      0.000      A 0.0023406244  3.374569e-06
  CH1    6      0.000      0.000      A 0.00606841  9.70225e-05
  CH2    6      0.000      0.000      A 0.0074684164  3.3965584e-05
  CH3    6      0.000      0.000      A 0.0096138025  2.6646244e-05
  CH4    6      0.000      0.000      A 0.01317904  3.4363044e-05
  CR1    6      0.000      0.000      A 0.0055130625  1.5116544e-05
   HC    1      0.000      0.000      A  8.464e-05  1.5129e-08
    H    1      0.000      0.000      A           0           0
  DUM    0      0.000      0.000      A           0           0
    S   16      0.000      0.000      A 0.0099840064  1.3075456e-05
 CU1+   29      0.000      0.000      A 0.0004182025  5.1251281e-09
 CU2+   29      0.000      0.000      A 0.0004182025  5.1251281e-09
   FE   26      0.000      0.000      A           0           0
 ZN2+   30      0.000      0.000      A 0.0004182025  9.4400656e-09
 MG2+   12      0.000      0.000      A 6.52864e-05  3.4082244e-09
 CA2+   20      0.000      0.000      A 0.00100489  4.9801249e-07
    P   15      0.000      0.000      A 0.01473796  2.2193521e-05
   AR   18      0.000      0.000      A 0.0062647225  9.847044e-06
    F    9      0.000      0.000      A 0.0011778624  7.6073284e-07
   CL   17      0.000      0.000      A 0.0087647044  1.5295921e-05
   BR   35      0.000      0.000      A 0.0011792356  6.5480464e-05
 CMET    5      0.000      0.000      A 0.0088755241  2.0852922e-05
 OMET    8      0.000      0.000      A 0.0022619536  1.505529e-06
  NA+   11      0.000      0.000      A 7.2063121e-05  2.1025e-08
  CL-   17      0.000      0.000      A 0.01380625  0.0001069156
 CCHL    6      0.000      0.000      A 0.0026308693  4.064256e-06
CLCHL   17      0.000      0.000      A 0.0083066819  1.3764842e-05
 HCHL    1      0.000      0.000      A 3.76996e-05  4.2999495e-09
SDMSO   16      0.000      0.000      A 0.010561673  2.149806e-05
CDMSO    6      0.000      0.000      A 0.0090514293  2.175756e-05
ODMSO    8      0.000      0.000      A 0.0022707131  7.5144626e-07
 CCL4    6      0.000      0.000      A 0.0026308693  7.5999462e-06
CLCL4   17      0.000      0.000      A 0.0076040144  1.2767758e-05
 CH2r    6      0.000      0.000      A 0.0073342096  2.8058209e-05
  CH0    6      0.000      0.000      A 0.0023970816  0.0002053489
   SI   14      0.000      0.000      A 0.01473796  2.2193521e-05
 MNH3    0      0.000      0.000      A  0.0            0.0
   MW    0      0.000      0.000      A  0.0            0.0
[ nonbond_params ]
; i j func c6 c12
      OM        O  1  0.0022619536  7.4149321e-07
      OA        O  1  0.0022619536  1.380375e-06
      OA       OM  1  0.0022619536  2.258907e-06
      OW        O  1  0.0024331696  1.825875e-06
      OW       OM  1  0.0024331696  2.987943e-06
      OW       OA  1  0.0024331696  1.991421e-06
       N        O  1  0.0023475616  2.185875e-06
; ...  et cetera
[ pairtypes ]
; i j func c6 c12
       O        O  1  0.0022619536  7.4149321e-07
      OM        O  1  0.0022619536  7.4149321e-07
      OM       OM  1  0.0022619536  7.4149321e-07
      OA        O  1  0.0022619536  9.687375e-07
      OA       OM  1  0.0022619536  9.687375e-07
      OA       OA  1  0.0022619536  1.265625e-06
      OW        O  1  0.0024331696  1.3975653e-06
      OW       OM  1  0.0024331696  1.3975653e-06
      OW       OA  1  0.0024331696  1.825875e-06
      OW       OW  1  0.0026173456  2.634129e-06
       N        O  1  0.0023475616  1.1202911e-06
; ...  et cetera
```

Christian Margreitter

Schlachthausgasse 35/11
1030, Wien
☎ 0043 680 10801632
✉ christian.margreitter@gmail.com

Education

since 2007	**Study of Informatics**, *University of Vienna*
since 2006	**Study of Molecular Biology**, *University of Vienna*
June 2005	**High School Diploma with Distinction**, *Bundesgymnasium Bludenz*

Experience

Apr 2011 - Jun 2012	**Diploma thesis**, *University of Vienna, MFPL*, Group of Bojan Zagrovic
Feb 2011 - Mar 2011	**Internship**, *ETH Zurich*, Group of Professor Matthias Peter Phospho-protein interaction study regarding *S. cerevisiae*'s Avo1p and Atg8p in the context of the *Cvt-pathway*
Oct 2010	**Internship**, *University of Vienna*, Group of Professor Gustav Ammerer Biochemical hands-on training on sample preparation and mass spectrometry-based analysis of phospho-proteins in *S. cerevisiae*
Aug 2005 - Aug 2006	**Alternative Civilian Service**, *Kolpinghaus*, Bregenz

Diploma thesis

Title *Vienna-PTM: Establishment of a server extending simulation capabilities of proteins by post-translational modifications*

Supervisor Bojan Zagrovic

Abstract Post-translational modifications (PTMs) of proteins have, over the last decades, been extensively investigated from a number of different aspects. They are involved in a manifold of critical processes in the cell, including signaling, regulation and localization control. PTMs make fast and predominantly reversible amino acid alteration possible, are often inter-dependent and build sometimes networks on their own. The impact on the physical-chemical properties of affected residues is often significant and potentially affecting the overall properties of the whole protein. However, *in silico* simulations of PTMs have been strongly neglected, especially considering their biological relevance. With increasing numbers of observed types and occurrences of PTMs, it seems therefore timely and important to include them in classical mechanical force fields used for biomolecular simulations. This work presents **Vienna-PTM**, a server designed to provide both a workflow for introducing post-translational modifications in protein PDB files as well as parameters for these modified amino acids. Thereby, the arsenal of possible building blocks is enriched from 20 to over 200 distinct residues, including common modifications such as phosphorylation, acetylation and methylation as well as a number of less widely used ones. All modifications are available for GROMOS force fields ffG45a3 and ffG54a7.

Other Projects

Bachelor Projects

C++ Implementation of an Radix-sort algorithm with variable frame length

Languages

German	**Native**
English	**Fluent**

Computer skills

Basic	JS, Perl, bash scripts
Intermediate	XML / DTD, UML, torque
Expert	C, C++, PHP, HTML, MySQL, LaTeX
Tools	gimp, inkscape, torque, GROMACS
Miscellaneous	Windows, Office, Linux

Qualities

Personal skills: Working independently, critical and analytical thinking, strength under pressure, reliable, assertive, communicative, eager to aquire new skills

Experience with: Programming and software engineering, computational simulations, yeast wet-lab research, purification and analysis of proteins and nucleic acids

References

Bojan ZAGROVIC Diploma thesis supervisor, *MFPL University of Vienna*