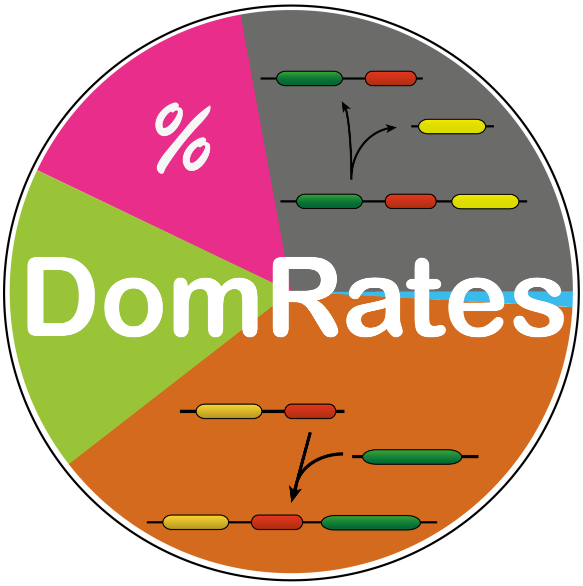

DomRates User Manual

Elias Dohmen

version 1.0.0
`http://domainworld.uni-muenster.de/`

Contents

1 Introduction

Modularity is an important keystone in molecular evolution and indispensible for evolutionary innovation. Protein domains as the modules of proteins can be reused in different molecular contexts and therefore rearrangements of domains can create a broad functional diversity with just a few mutational steps. **DomRates** infers these rearrangement events of protein domains for a given phylogeny and calculates the frequency of the related events.

This User Manual tries to be as detailed and all-encompassing as possible to answer all questions one might have when analyzing data with **DomRates**. Still we can't guarantee that everything is covered in the minutest details or that some bugs avoided us. If you have any questions or feedback, please contact us at domainworld@uni-muenster.de

2 Installation

DomRates has some dependencies, although we try to keep these to a minimum. The following existing packages or programs are needed to run DomRates:

- cmake (`https://cmake.org`)
- compiler supporting C++11 (e.g. g++)
- boost (`http://www.boost.org/`)
- BioSeqDataLib (see download instructions below)
  optional, but recommended:
- git (`https://git-scm.com/`)

In most Linux distributions (e.g. Ubuntu) it should be possible to install these dependencies using the package manager. You can try in Ubuntu for example the following commands to install the dependencies:

```
$ sudo apt-get install cmake
$ sudo apt-get install g++
$ sudo apt-get install libboost-all-dev
```

There are two ways to download *DomRates*. Either you can download it using git or manually from the website. Both ways are described below, although we recomment to use git.

2.1 Installation with git

You can use git to clone the repository and download BioSeqDataLibrary as a submodule with the following commands:

```
$ git clone https://ebbgit.uni-muenster.de/domainWorld/DomRates.git
$ cd DomRates
$ git submodule init
$ git submodule update
```

Inside the source folder a build directory is needed in which the code will be compiled. CMake is used to find all the needed requirements of the library. You can simply compile it using the following commands:

```
$ mkdir build
$ cd build
$ cmake ..
$ make
```

2.1.1 Updating with git

Sometimes it will be necessary to update *DomRates* or the underlying BioSeqDataLib, either because it contains some new features or because we unfortunately had a bug somewhere that we have fixed. You can simply use git to update your code. Just change into the DomRates directory and type the following commands:

```
$ git pull
$ git submodule foreach git pull origin master
```

In the following, you have to change to the build directory again and run the following commands from the installation section:

```
$ cmake ..
$ make
```

DomRates and the BioSeqDataLib are now updated successfully.

2.2 Manual installation without git

You can download the compressed source code from our gitlab webpage:
`https://ebbgit.uni-muenster.de/domainWorld/DomRates`

Additionally you will have to download the BioSeqDataLib
(`https://ebbgit.uni-muenster.de/domainWorld/BioSeqDataLib`)
and uncompress it in the libs directory of *DomRates*.

Inside the source folder a build directory is needed in which the code will be compiled. CMake is used to find all the needed requirements of the library. You can simply compile it using the following commands:

```
$ mkdir build
$ cd build
$ cmake ..
$ make
```

2.2.1 Updating without git

If you downloaded the code without git you will have to download the latest version and replace the old one with it. In the following you have to follow the steps described in the installation section above again.

2.3 Python and ETE3 for visualizations

If you want to use the visualization script for your results that is included in DomRates, you will need python (either in version 2 or 3) and ete3.

Details and downloads for python you find here: `https://www.python.org/`

Downloads and install instructions for ete3 are available here: `http://etetoolkit.org/download/`

With the following command you can test if the installation was successful and get further information about how to use the script:

```
$ python visualization_domrates_tree.py --help
```

3 Required Data / File Formats

To analyse your species set with *DomRates* you need the following data in the described file formats:

1. Phylogenetic tree (bifurcating and in newick format)

example_tree.nwk

```
(((B_tau,(P_tro,P_abe)),(G_gal,M_gal)),Outgroup);
```

2. Proteome domain annotations (annotated either with PfamScan or InterProScan)

example_pfam_file.dom

```
# pfam_scan.pl, run at Mon Jan 23 11:03:28 2017
# ...
# <seq id> <alignment start> <alignment end> <envelope start> <envelope end> <hmm acc> <hmm name> <type> <hmm start> <hmm end> <hmm length> <bit score> <E-value> <significance> <clan>
Seq1    5  199    2  199 PF11705.7 RNA_pol Family   2 229 229 127.0 1.2e-36 1 No_clan
Seq2  130  195  130  197 PF04505.11 CD225  Family   1  66  68  56.7 1.8e-15 1 No_clan
Seq3  397  470  397  471 PF11515.7  Cul7   Family   1  77  78 110.4 3.6e-32 1 CL0049
Seq3 1198 1315 1176 1336 PF03256.15 ANAPC1 Family  42 162 185  41.1 1.4e-10 1 CL0202
Seq3 1320 1822 1288 1838 PF00888.21 Cullin Family 172 600 618 106.6 1.3e-30 1 No_clan
...
```

We recommend to filter the studied sequence data for different isoforms and just to keep one each time (for example just the longest one) to avoid multiple isoforms of the same protein influencing the rearrangement analysis. For this purpose you can use for example the tool *isoformCleaner* (`https://ebbgit.uni-muenster.de/domainWorld/dw-helper`).

```
Note: The annotation files need to have the same names as the leaves in your tree.
    By default DomRates searches for files with the file extension '.dom'
```

For details about how to annotate your sequence data with either PfamScan or InterProScan please visit their homepages (`http://pfam.xfam.org/` or `https://www.ebi.ac.uk/interpro/`). Standalone annotation tools can be downloaded for PfamScan (`ftp://ftp.ebi.ac.uk/pub/databases/Pfam/Tools/`) and InterProScan (`https://www.ebi.ac.uk/interpro/download.html`).

3.1 Data preparation

In the following the preparation of an example proteome for use with DomRates is shown, including isoform cleaning, domain annotation and quality control. The example proteome we use in this example will be the *Danio rerio* proteome, available for download from the ensembl database via this link `ftp://ftp.ensembl.org/pub/release-95/fasta/danio_rerio/pep/Danio_rerio.GRCz11.pep.all.fa.gz`. Once the file has been extracted you should have the fasta file containing all protein sequences for *Danio rerio*. As mentioned above smaller isoforms should be removed from your data to avoid interpreting isoform differences as domain rearrangements. For this purpose we can download the isoformCleaner here `https://ebbgit.uni-muenster.de/domainWorld/dw-helper` and after installing it (see instructions in the dw-helper manual) run it on the proteome file with the following command:

```
$ isoformCleaner -i Danio_rerio.GRCz11.pep.all.fa -o Danio_rerio.GRCz11.pep.all.iso.fa -p gene
```

This should result in around 30,000 sequences in the output file after isoform cleaning instead of around 50,000 sequences in the original input file.

We can now annotate this isoform free proteome with protein domains. For this purpose we can use for example the PfamScan script (`ftp://ftp.ebi.ac.uk/pub/databases/Pfam/Tools/`) and run it on our proteome with the following command (for more details check out the PfamScan documentation):

```
$ pfam_scan.pl -fasta Danio_rerio.GRCz11.pep.all.iso.fa -dir /pathto/pfam/
 -outfile Danio_rerio.GRCz11.pep.all.iso.fa.dom
```

The resulting annotation file can be used in DomRates directly. One last step, however, would be to check the quality of your proteome. Comparing species with proteomes of very different quality can lead to over- or underestimation of several rearrangement events and introduce a bias that could be avoided if species of similar and high quality are used. The DOGMA webserver (`https://domainworld-services.uni-muenster.de/dogma/`) offers a quick way to check the quality of the proteome. Simply select the annotation file produced before (*Danio_rerio.GRCz11.pep.all.iso.fa.dom*) and select as a core set a phylogenetic clade your organism belongs to. In this case this would be for example 'vertebrates'. Finally, you can hit the 'submit job' button at the bottom and you should get the result (completeness score %) after a few seconds. To do this for multiple species it could be useful to download and use the standalone version of DOGMA (`https://domainworld.uni-muenster.de/programs/dogma/`).

4 Running DomRates - Basic Commands

After the installation of **DomRates** as described in section 2, the following commands are available. A short description of the allowed parameters of **DomRates** can be accessed via the following command:

```
$ domRates --help
```

```
Note: To run DomRates like this you have to call it from the build directory or
     put DomRates into the PATH-Variable. Otherwise please give the full path of
     DomRates to run it.
```

A basic way of analysing your data with **DomRates**, if you just want to know the frequencies of the different rearrangement events in your data set, would be:

```
$ ./domRates -t tree.nwk -a /dir/with/annotation-files/ -g outgroup -o rearrangement_freqs.txt
```

tree.nwk: This should be a tree file in newick format (for file format see section 3)

/dir/with/annotation-files/: path to directory with annotation files (for file format see section 3)

outgroup: the name of the outgroup as labeled in the tree

rearrangement_freqs.txt: the name of the output file

4.1 The -s / --statistics parameter

For a more detailed output with additional statistics that is also needed to visualise the findings like described below (see section 5), you can run DomRates with the following parameter:

```
$ ./domRates -t tree.nwk -a /dir/with/annotation-files/ -g outgroup -o rearrangement_freqs.txt
 -s stats_file.txt
```

tree.nwk: This should be a tree file in newick format (for file format see section 3)

/dir/with/annotation-files/: path to directory with annotation files (for file format see section 3)

outgroup: the name of the outgroup as labeled in the tree

rearrangement_freqs.txt: the name of the output file

stats_file.txt: the name of the output file with additional statistics

When the -s parameter is specified, two files with statistics are created. The first file created has the name as provided with the -s parameter (in this example *stats_file.txt*), while the second file has the same output name with the appendix ”_epd” (in this example *stats_file_epd.txt*). For a detailed description of the content of these files see section 5.

4.2 The -d / --detailed parameter

DomRates' main purpose is it to reconstruct and explain all arrangements that changed over the phylogenetic tree in any way. Therefore, it lists only arrangements in the previous mentioned output files (-o and -s parameters), which changed and are explicitly explainable. If you are additionally interested which arrangements are conserved in your data set and do not change at the different nodes of your tree or for which arrangements multiple solutions were found, you can get that information by using the following command:

```
$ ./domRates -t tree.nwk -a /dir/with/annotation-files/ -g outgroup -o rearrangement_freqs.txt -d
```

tree.nwk: This should be a tree file in newick format (for file format see section 3)

/dir/with/annotation-files/: path to directory with annotation files (for file format see section 3)

outgroup: the name of the outgroup as labeled in the tree

rearrangement_freqs.txt: the name of the output file

If next to the -o parameter also the -s parameter is specified (see section 4.1) these statistic files will also contain information about maintained arrangements and ambiguous or complex solutions.

```
Note: Listing all maintained arrangements and not unique solutions with the −d
    parameter can heavily increase the file size of the statistic files dependent
    on the size of the data set.
```

4.3 The -e / --ending parameter

Species in the phylogenetic tree are matched to the corresponding annotation files (provided via the -a parameter) by adding the file extension (default: '.dom') to the leave names.

If your tree contains for example as leaf names *C_elegans* and *D_melanogaster*, there have to be annotation files in the provided annotation directory with the file names *C_elegans.dom* and *D_melanogaster.dom*. You can adapt the file extension DomRates scans for by using the -e parameter:

```
$ ./domRates -t tree.nwk -a /dir/with/annotation-files/ -g outgroup -o rearrangement_freqs.txt
 -e .pep.all_iso.pfam
```

tree.nwk: This should be a tree file in newick format (for file format see section 3)

/dir/with/annotation-files/: path to directory with annotation files (for file format see section 3)

outgroup: the name of the outgroup as labeled in the tree

rearrangement_freqs.txt: the name of the output file

.pep.all_iso.pfam: the file extension of your annotation files

In this example for the species names in the tree *C_elegans* and *Drosophila-melanogaster* the annotation directory would be scanned for the annotation files *C_elegans.pep.all_iso.pfam* and *Drosophila-melanogaster.pep.all_iso.pfam*.

4.4 The -p / --threads parameter

To speed up your DomRates calculations you can specify how many threads of your CPU can be used for it. To run DomRates with 8 threads in parallel for example you can use the following command:

```
$ ./domRates -t tree.nwk -a /dir/with/annotation-files/ -g outgroup -o rearrangement_freqs.txt -p 8
```

tree.nwk: This should be a tree file in newick format (for file format see section 3)

/dir/with/annotation-files/: path to directory with annotation files (for file format see section 3)

outgroup: the name of the outgroup as labeled in the tree

rearrangement_freqs.txt: the name of the output file

8: number of threads to use for the calculation (check before how many cores/threads are available)

The use of multiple threads can shorten the runtime massively, however to use more threads than nodes exist in the provided tree (sum of all leaves and inner nodes) does not speed up DomRates anymore. To be able to run DomRates with multiple threads you need a compiler that supports OpenMP (what should be the case for most of the current compilers).

4.5 The -n / --node parameter

If you are interested in statistics about one specific node in your tree, you can use the -n parameter in combination with the -s parameter. If two species names devided by ’:’ are provided, all arrangements involved in rearrangement events at the node representing the last common ancestor of both species will be listed in the statistics file in a separate section. This parameter is just usable if -s parameter is set. To get for example information about the last common ancestor of *D_melanogaster* and *C_elegans* in your data set you can use the following command:

```
$ ./domRates -t tree.nwk -a /dir/with/annotation-files/ -g outgroup -o rearrangement_freqs.txt
 -s stats_file.txt -n D_melanogaster:C_elegans
```

tree.nwk: This should be a tree file in newick format (see section 3)
/dir/with/annotation-files/: path to directory with annotation files (see section 3)
outgroup: the name of the outgroup as labeled in the tree
rearrangement_freqs.txt: the name of the output file
stats_file.txt: the name of the output file with additional statistics
D_melanogaster:C_elegans: tip labels of the two species of which last common ancestor information should be stored in the statistics file

The section in the statistics file will be located at the end of the file and starts with a line like this:

stats_file.txt

```
# Events per domain arrangement for last common ancestor of D_melanogaster:C_elegans
...
```

5 Results - Interpretation and Visualization

This section should give a detailed description of how to interpret the output **DomRates** produces and how to filter and visualize the findings according to your needs.

5.1 Event types

The possibly most important information is which different rearrangement events, including emergence and loss of domains, are covered by **DomRates**. Six different event types can be distinguished, such as Fusion, Fission, Terminal Loss/Emergence and Single Domain Loss/Emergence as shown in Figure 1.

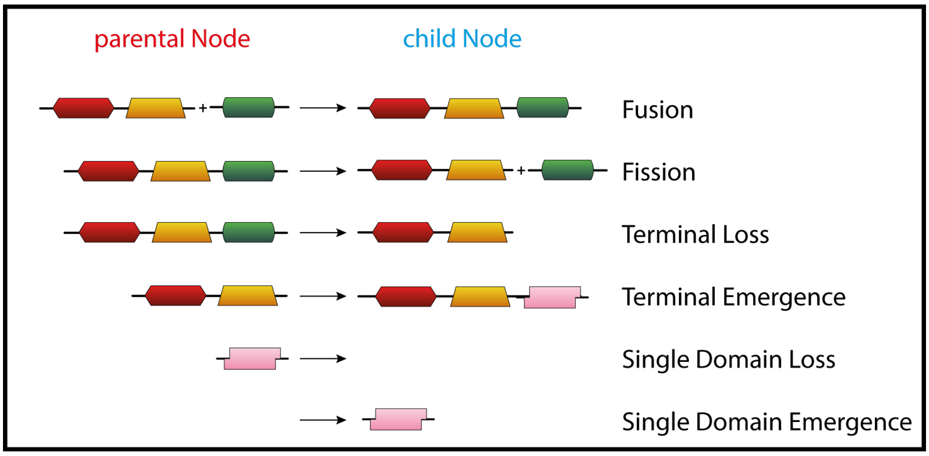

Figure 1: **Event types.** Six different event types are distinguished based on the underlying mechanisms leading to that type of an event.

The domain arrangement content of every node in the given phylogenetic tree is reconstructed and for every arrangement that exist at a certain node, but was not present in the parental node, it is checked if one of the here shown events can explain the new arrangement. The same applies to a missing single domain, that was existent in the parental node.

5.2 Solution types

Based on the different event types it is possible to distinguish four different solution types, dependent on how many solutions could explain a new arrangement. The different solution types can be seen in Figure 2.

If just one event was found that could explain a new arrangement, it is considered an exact solution. Furthermore, it is possible that more than one event can possibly explain a new arrangement. If all found events belong to the same event type (e.g. all are fusions) it is considered a non-ambiguous solution and considered with all exact solutions in DomRates, since the underlying event type can be inferred with certainty. Different event types that could explain a new arrangement are considered

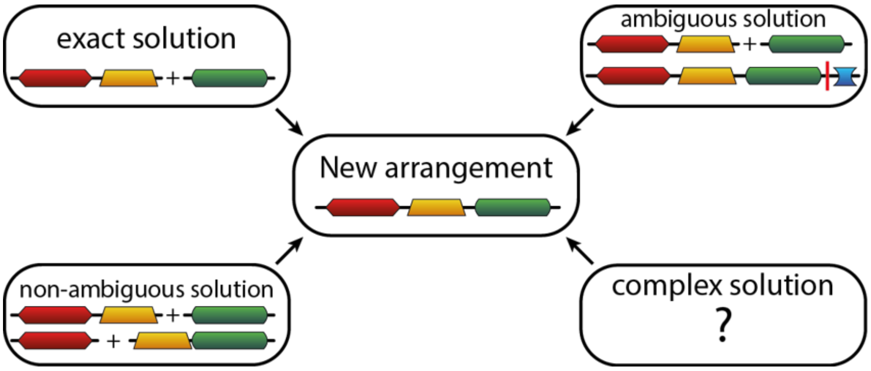

Figure 2: **Solution types.** Dependent on the amount and type of events that can explain the new arrangement four different solution types can be distinguished.

as an ambiguous solution. The last case - a complex solution - describes a missing explanation for a new arrangement. This can be the case if the resolution of the tree is not good enough to infer single-step events due to missing steps in between which are represented in the dataset as a chain of multiple events. Another explanation can be a rare event that led to a new arrangement, which was not considered in **DomRates**.

5.3 The main output file

The main output file **DomRates** produces is given via the -o / --output parameter. It looks like in the following example:

```
# DomRates version 1.0.0 at Wed Mar  6 23:35:19 2019
# domRates -t example_tree.nwk -a annotations/ -g Saccharomyces_cerevisiae -e .dom -o rearrangement_freqs.txt -s stats.txt -n  -d 0 -p 1
# Solution types
Exact solutions: 2056
Non-ambiguous solutions: 213
Ambiguous solutions: 291
Complex solutions: 1853

Exact and non-ambiguous solutions: 2269

# Event types
Fusions: 801
Fissions: 390
Terminal Loss: 416
Terminal Emergences: 82
Single Domain Losses: 358
Single Domain Emergences: 222

# Event rates
Fusion rate: 35.30%
Fission rate: 17.19%
Terminal Loss rate: 18.33%
Terminal Emergence rate: 3.61%
Single Domain Loss rate: 15.78%
Single Domain Emergence rate: 9.78%
```

This file gives an overview of how many arrangement changes have been found. The first block (line 1 to 4) lists the number of events belonging to different solution types (see section 5.2). The more exact solutions have been found, the better. Together with the non-ambiguous solutions they are the only ones accounting for the inferred event types shown in the second block of the file (line 6 to 11) in total numbers (the total number of considered events is summed up in line 13). A detailed description of the different event types can be found in section 5.1. The third block of the file (line 15 to 20) lists the frequency of every event type in the data set in percent, instead of total numbers. With this percentage it is easily visible to what extend the different event types account for changes in your data set.

Usually, Fusion is one of the most frequent event types. Both emergence types should be usually very rare. A high overall emergence frequency is often a hint for issues with the data set (see for example section 6.3). Loss of complete domains or part of arrangements can be a very important and frequent or very rare evolutionary mechanism dependent on the clade and genomic properties of the studied species.

5.4 The optional statistic files

For further analyses more information than just the overall frequency of event types in the dataset can be useful. Therefore, the -s /--statistics parameter can be used to get two files with additional information about domains and their distribution across the phylogeny with all events they are involved in. The files are tab separated and this way can be analyzed and filtered using spreadsheet software or by automated processing scripts. This file is also needed for visualization of the data with the provided script (see section 5.5).

The first file with exactly the same name as provided via the -s parameter should look like the following one:

stats.txt:

```
1   # Number of events per node.
2   # Node ID       #Fusions        #Fissions       #TerminalLosses #TerminalEmergences
            #SingleDomainLosses     #SingleDomainEmergences
3   0       0       0       0       0       0       0
4   1       0       0       0       0       0       0
5   2       161     69      86      7       42      14
6   3       92      37      36      26      3       91
7   4       113     73      81      5       168     13
8   5       98      35      26      21      5       41
9   6       63      61      59      6       53      4
10  7       53      12      13      6       3       16
11  8       134     47      51      7       63      32
12  9       87      56      64      4       21      11
13  10      0       0       0       0       0       0
```

Here are listed for every single node in the tree the number of total events for every event type. This section starts every time with exactly the lines as shown in line 1 and 2 of the example file. Line 2 gives information about which column lists the number of events for which event type (for information about event types see section 5.1). Getting information about which node in the tree was mapped to what ID can be found further down in section 5.5.

A second section can optionally appear in this file, when the -n parameter is used (see section 4.5). It starts with two comment lines and has the following general structure:

```
# Events per domain arrangement for last common ancestor of Loxodonta_africana: Bos_taurus.
# Node-ID  solution type  event type  new arrangement at current node  arrangement at parental node
    5     exact solution   fission     PF00051 PF01822 | PF00431   PF00051 PF01822 PF00431
    5     exact solution   fission     PF00096 PF13465 | PF00096   PF00096 PF13465 PF00096
    5     exact solution   fission     PF00130 PF16664 PF14604 | PF07653   PF00130 PF16664 PF14604 PF07653
...
```

The information from this section shows all events that happened at the node that represents the last common ancestor of both given species. The same information is also included in the second statistics file, but there you need to find the node by its ID (see section 5.5). A more detailed description can be found in the next paragraph.

The second statistics file matches the name provided with the -s parameter with the addition *_eps. It looks like the following one:

stats_epd.txt:

```
# Node-ID    solution type    event type    new arrangement at current node  arrangement at parental node
    2       exact solution     fission       PF00018 PF14604 PF00018 | PF00017   PF00018 PF14604 PF00018 PF00017
    2       exact solution     fusion        PF00001 PF00091 PF03953   PF00001 + PF00091 PF03953
    2       exact solution   terminal loss   PF00004 PF08519 PF00533   PF00004 PF08519
...
```

This file contains, for each node in the tree, all arrangements with their corresponding solution and event type. The first line gives information about what to find in each column. If the -d / --detailed parameter was used (see section 4.2) this file contains also maintained arrangements, which did not change as well as arrangements that are part of an ambiguous or complex solution.

Further information about the domains by their accessions can be looked up dependent on the annotation used on the pfam homepage (`http://pfam.xfam.org/`) or the interpro homepage (`https://www.ebi.ac.uk/interpro/`).

The meaning of the node IDs listed in the first column is described in the following section.

5.5 Visualizing results and mapping node IDs

The node IDs given in the statistic files can be easily assigned to their nodes in the phylogenetic tree with the provided Python script in the DomRates/src/ directory (for installation see section 2.3).

With the following command a pdf (*tree_ids.pdf*) can be created showing the phylogenetic tree with all node IDs as used in the statistic files:

```
$ python visualization_domrates_tree.py -t tree.nwk -y tree_ids
```

The provided *tree.nwk* file should be the same file as used for the analysis with DomRates. The result should look similar to Figure 3.

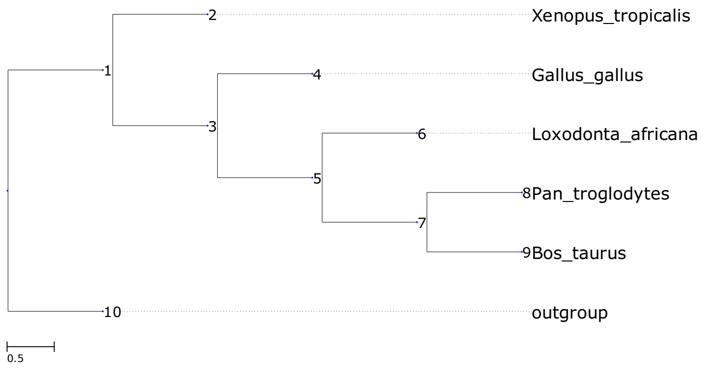

Figure 3: **Tree with node IDs.**

The script can furthermore be used to visualize the results, for example with the following command (see also section 6):

```
$ python visualization_domrates_tree.py -t tree.nwk -s stats.txt -o tree_event_distr
```

The file *tree.nwk* should be the same as used for the DomRates analysis and *stats.txt* should be the statistics file received from that analysis.

The tree can then look like in Figure 4 and shows the amount of different event types inferred at every node in percentage represented as a pie chart and the total number in digit representation right to the pie chart. The legend for the color coding of all event types is given at the top.

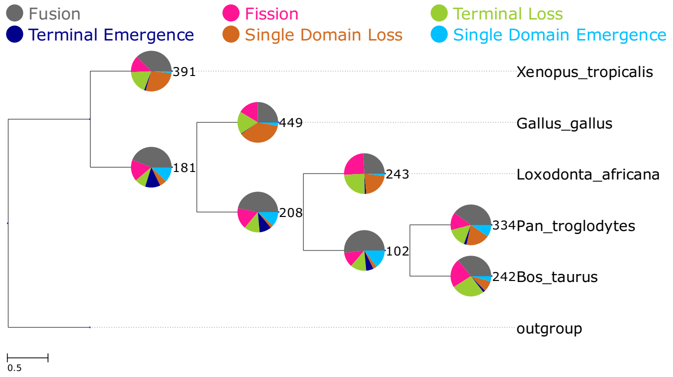

Figure 4: **Tree visualization for distribution of all events.**

For further details please use the following command or see section 6:

```
$ python visualization_domrates_tree.py --help
```

6 Analyzing an Example Data Set

The following example shows how to analyse a data set with **DomRates** step by step. If not done yet, please install DomRates first before you start this example. For installation instructions see section 2. Please change to the subdirectory ”examples” in the DomRates directory to run the commands shown in this example. If you want to compare your results or you have trouble running some commands, you find all files that are produced in this example in the directory ”results” located in the ”examples” directory.

6.1 Running DomRates on the data set

All files necessary to run **DomRates** with this example are already provided. The phylogenetic trees have the extension *.nwk, all files with the extension *.dom are proteomes, already annotated with PfamScan (for help with annotating sequence files or supported tree formats see section 3).

```
Note: Please be aware that the annotation files in this example do not represent
    the full proteome of the species and are edited to give a good overview. They
    should not be used for other purposes or analyses.
```

The species we want to analyse regarding their protein domain arrangement changes are the following: *Xenoput tropicalis* the Western clawed frog, *Gallus gallus* the Red junglefowl, *Loxodonta africana* the African bush elephant, *Pan troglodytes* the Common chimpanzee, *Bos taurus* the Cattle and additionally an outgroup.

DomRates can infer all these changes automatically if we just provide a phylogenetic tree and annotated proteomes of the species included. We can do this with the following command:

```
$ ../build/domRates -t example_tree.nwk -a annotations/ -g Saccharomyces_cerevisiae
 -o rearrangement_freqs.txt -s stats.txt
```

If your compiler supports OpenMP and you want **DomRates** to run faster, you can use additionally the -p parameter followed by the number of threads to use (see section 4.4).

6.2 Results

Three files are created by **DomRates** this way: *rearrangement_freqs.txt*, *stats.txt* and *stats_epd.txt*. Let’s have a look at the first file, it should look like this:

rearrangement_freqs.txt:

```
# DomRates version 1.0.0 at Wed Mar  6 23:35:19 2019
# domRates -t example_tree.nwk -a annotations/ -g Saccharomyces_cerevisiae -e .dom -o rearrangement_freqs.txt -s stats.txt -n  -d 0 -p 1
# Solution types
Exact solutions: 2056
Non-ambiguous solutions: 213
Ambiguous solutions: 291
Complex solutions: 1853

Exact and non-ambiguous solutions: 2269

# Event types
Fusions: 801
Fissions: 390
Terminal Loss: 416
Terminal Emergences: 82
Single Domain Losses: 358
Single Domain Emergences: 222

# Event rates
Fusion rate: 35.30%
Fission rate: 17.19%
Terminal Loss rate: 18.33%
Terminal Emergence rate: 3.61%
Single Domain Loss rate: 15.78%
Single Domain Emergence rate: 9.78%
```

In section 5 you find a detailed description of the solution types (section 5.2) and event types (section 5.1) listed here and how to interpret these. In the first block we see that we have quite a lot of Complex solutions compared to the Exact solutions, which could be maybe reduced if we would increase the resolution of our tree. We also see a quite high Single Domain Emergence rate of 9.7%, which could be due to the choice of our outgroup. Generally, emergence of a new domain is a very rare event and a high percentage is either a signal of important evolutionary changes or an issue with our dataset. We will check that later.

First we want to see how these events distribute over the phylogenetic tree and if we see potential hot-spots for some events that are interesting to analyse further.

The second file contains information about the number of events for every single node in our phylogenetic tree and should look like this:

stats.txt:

```
# Number of events per node.
# Node ID       #Fusions        #Fissions       #TerminalLosses # TerminalEmergences     #SingleDomainLosses     #SingleDomainEmergences
0       0       0       0       0       0       0
1       0       0       0       0       0       0
2       161     69      86      7       42      14
3       92      37      36      26      3       91
4       113     73      81      5       168     13
5       98      35      26      21      5       41
6       63      61      59      6       53      4
7       53      12      13      6       3       16
8       134     47      51      7       63      32
9       87      56      64      4       21      11
10      0       0       0       0       0       0
```

A more detailed description about this file and how to interpret it can be found in section 5.4. We want to use this file now to visualize our findings. For this purpose there is a python script in the src/ directory of the DomRates folder. You need a python installation and the package ete3 on your system in order to run it (more information in section 2.3).

You can run it with the following command to get a visualization of the data:

```
$ python ../src/visualization_domrates_tree.py -t example_tree.nwk -s stats.txt -l
 -o tree_event_distr
```

This creates the file *tree_event_distr.pdf*, which should look like Figure 5.

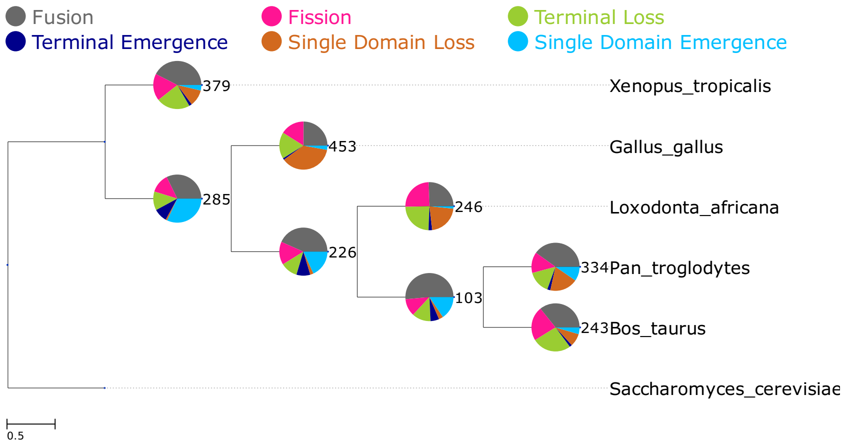

Figure 5: **Visualization of distribution of all events across the phylogenetic tree.**

A detailed description on how to interpret this figure can be found in section 5.5. Interesting is here for example the high percentage of Single Domain Emergences at the node ancestral to all birds and mammals, while a lot of Single Domain Losses seem to take place at the branch leading to *Gallus gallus*. The highest number of events is also located at this node.

If we want to take a closer look at the Single Domain Emergences, we can rerun the visualization script with the following parameters:

```
$ python ../src/visualization_domrates_tree.py -t example_tree.nwk -s stats.txt
 -o tree_sdemergences -e singleDomainEmergences -c 0.5
```

-e singleDomainEmergences: determines the event type that should be plotted

-c 0.5: specifies a scaling factor to let nodes with more events appear bigger

This creates the file *tree_sdemergences.pdf*, which shows the distribution of Single Domain Emergences across our phylogeny:

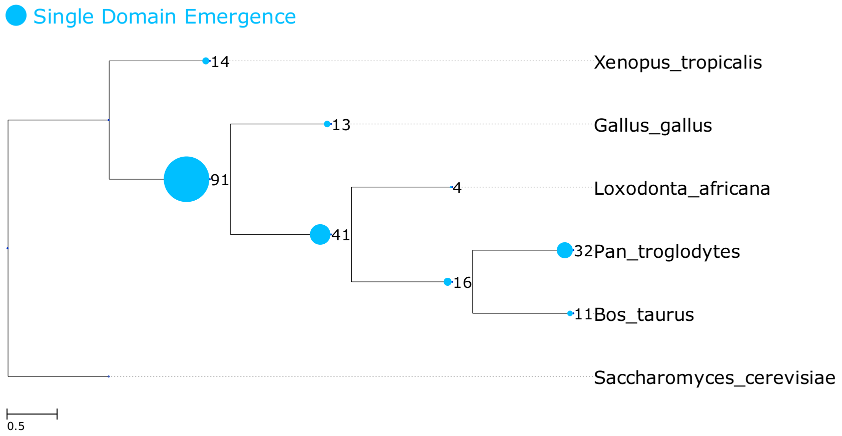

Figure 6: **Single Domain Emergences across the phylogenetic tree.**

The most Single Domain Emergences cluster at a very deep node in the tree, including all species except *Xenopus tropicalis*.

6.3 Evaluation of data bias - the choice of an outgroup

To validate if the results found above are really a biological finding pointing to differences between all analyzed species and *Xenopus tropicalis* and not a bias by our choice of an outgroup, we should rerun the analysis with a modified outgroup. We can combine the proteome annotations of multiple species to take them all into account as an outgroup instead of just picking one single organism. For this purpose, we will use next to *Saccharomyces cerevisiae* also *Drosophila melanogaster* and *Caenorhabditis elegans*. Their proteome annotations are already combined in the file *outgroup.dom* and a tree replacing *Saccharomyces cerevisiae* with 'outgroup' is stored in the file *example_tree_wog.nwk*.

Run **DomRates** with this modified outgroup:

```
$ ../build/domRates -t example_tree_wog.nwk -a annotations/ -g outgroup
 -o rearrangement_freqs_outgroup.txt -s stats_outgroup.txt
```

When we compare now the new results in *rearrangement_freqs_outgroup.txt* with the old ones shown above, we find a slightly lower number of total events (2152 vs. 2271). While the Single Domain Emergence rate dropped (5.81% vs 9.78%), the Single Domain Loss rate was increased (20.07% vs 15.76%).

Running the visualization script with the new results shows us the changes that happened in the tree:

```
$ python ../src/visualization_domrates_tree.py -t example_tree_wog.nwk -s stats_outgroup.txt -l
 -o tree_outgroup
```

The file *tree_outgroup.pdf* looks now like this:

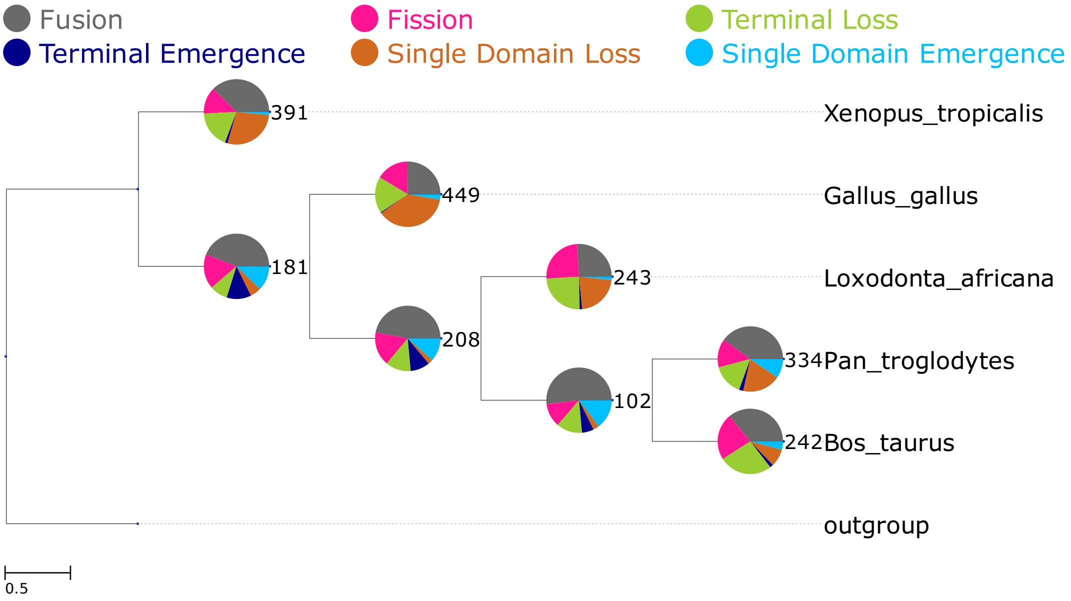

Figure 7: **Visualization of distribution of all events across the phylogenetic tree.**

Xenopus tropicalis gained much more Single Domain Losses, while most of the Single Domain Emergences at the ancestral node leading to all other species disappeared. Here it becomes clear that the right choice of an outgroup is important to infer correct domain changes. The additional emergences detected before were mainly caused by a lack of these domains in the outgroup *Saccharomyces cerevisiae*, although they existed already in earlier ancestors and different clades. The loss of multiple domains in *Xenopus tropicalis* was for the same reason not visible before, since these domains were also not present in the chosen outgroup *Saccharomyces cerevisiae*.

6.4 A closer look at new functionality

Finally, we want to have a closer look at the new domains which emerged at some of the nodes and see if we can associate them with specific functions. A visualization of just the emergences based on the calculations with the new outgroup shows 24 Single Domain Emergences at the root of all mammal species in our data set.

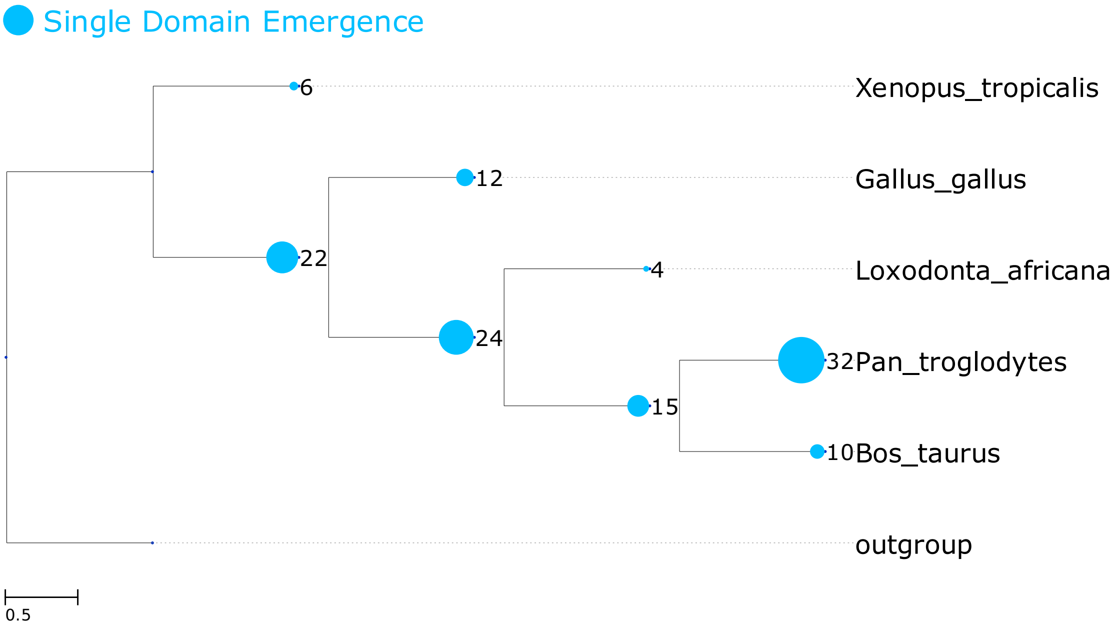

Figure 8: **Single Domain Emergences across the phylogenetic tree.**

To be able to have a more detailed look at these ones, we need to know the ID of the node we are interested in. For this purpose we can simply run the visualization script with the -y parameter via the following command:

```
$ python ../src/visualization_domrates_tree.py -t example_tree_wog.nwk -y tree_ids
```

We can see in the resulting *tree_ids.pdf* that the node ID of the node we are interested in is 5:

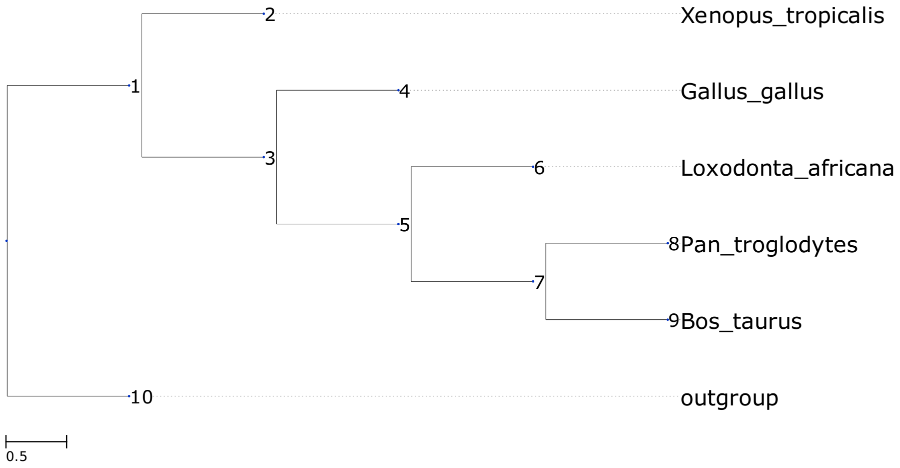

Figure 9: **Tree with node IDs.**

The file *stats_outgroup_epd.txt* contains all detailed information about the different events with their according node numbers (for more information about this file see section 5.4). To get all Single Domain Emergences (which are exact solutions) for node 5, we can use for example a simple grep command to pick just the relevant lines from the file:

```
$ grep -P "5\texact solution\tsingle domain emergence" stats_outgroup_epd.txt
```

The output we get is a list of all Single Domain Emergences at node 5:

```
1 5       exact solution  single domain emergence PF00260
2 5       exact solution  single domain emergence PF00363
3 5       exact solution  single domain emergence PF00414
4 5       exact solution  single domain emergence PF00715
5 5       exact solution  single domain emergence PF00997
6 5       exact solution  single domain emergence PF01099
7 ...
```

The fourth column contains the Pfam accessions for the domains that emerge at node 5 and can be checked for example via the Pfam homepage (`http://pfam.xfam.org/`).

When we have a look at the domains listed in line 4 and 5 (PF00715 and PF00997) we find the first one to be associated to Interleukin2 and the second one to Kappa casein. Interleukins as important parts of the mammalian immune system and Kappa casein as a mammalian milk protein can give some insight already into functional adaptations that happened at the root of all mammals. Emerging domains provided here new functionalities that were keystones for the success of mammals in evolutionary history.

With analyses as described in this example we can try to trace some important evolutionary adaptations and rearrangement events that influenced protein innovation.