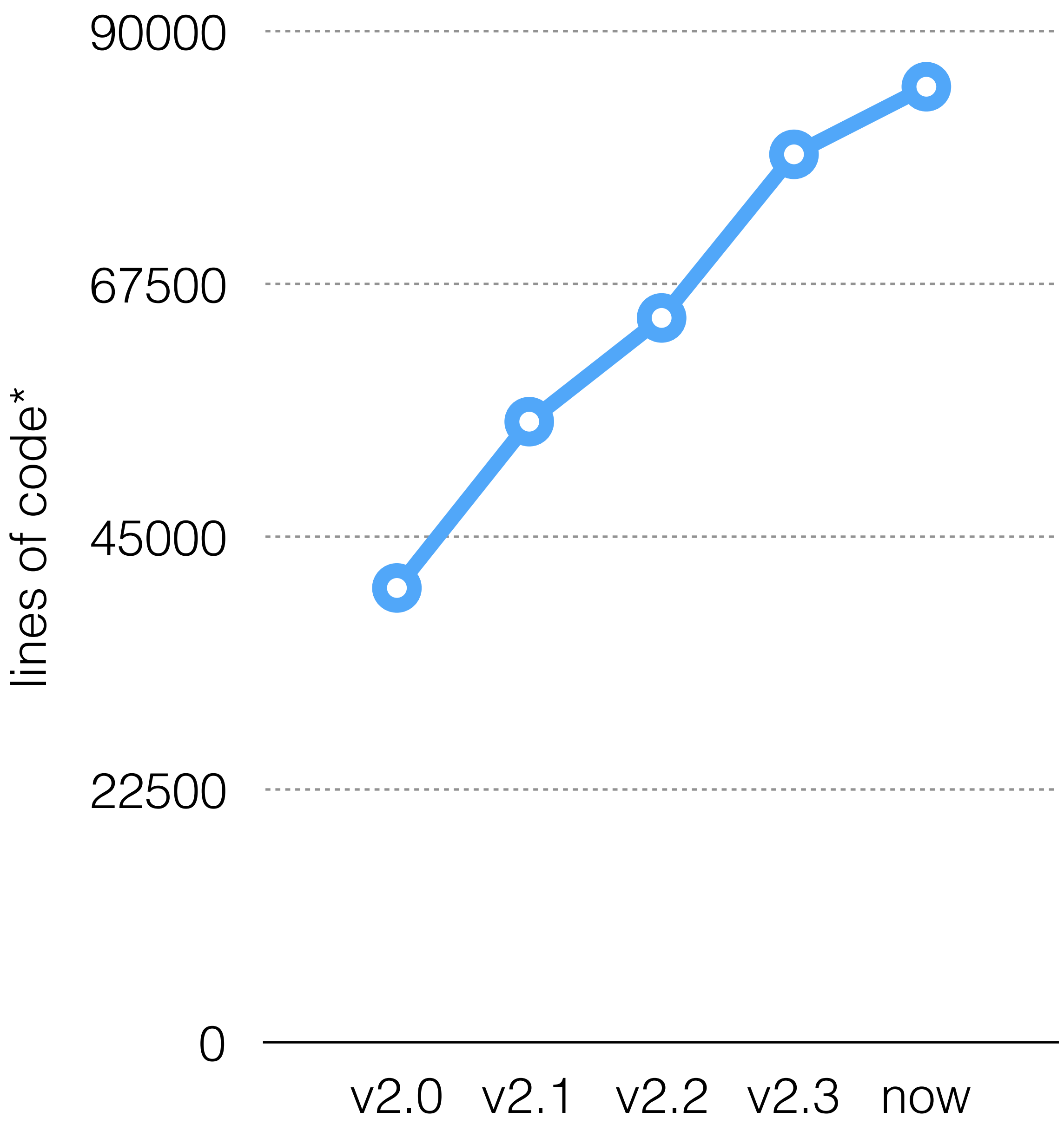
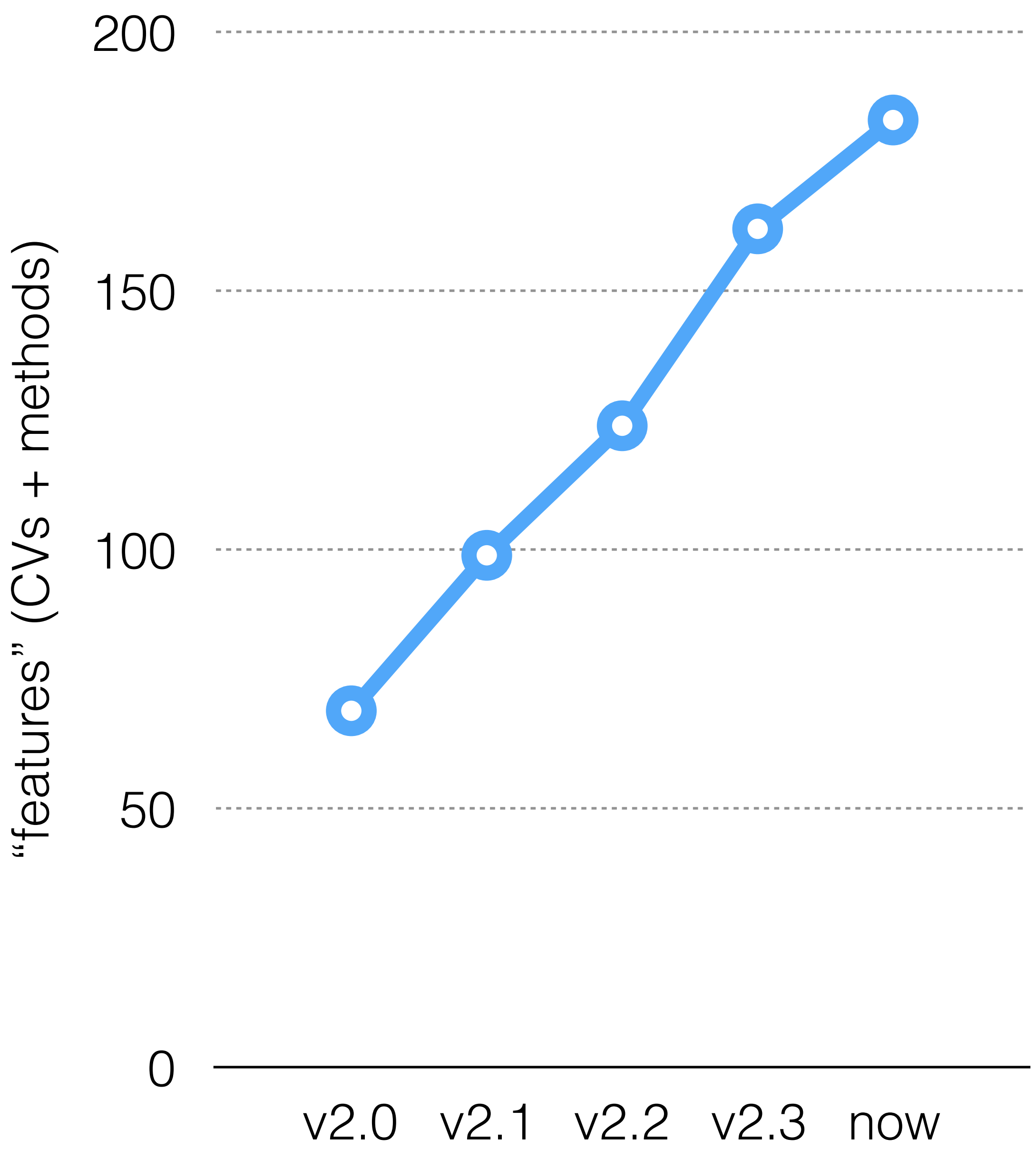
PLUMED 2
Contributing to PLUMED

Gareth Tribello
Massimiliano Bonomi
Carlo Camilloni
Giovanni Bussi

http://www.plumed.org

PLUMED is growing quickly





*excluding lapack/blas/VMD plugins

Many ways to contribute

Participating on the mailing list (with questions and answers)

Sharing information about how you used PLUMED*

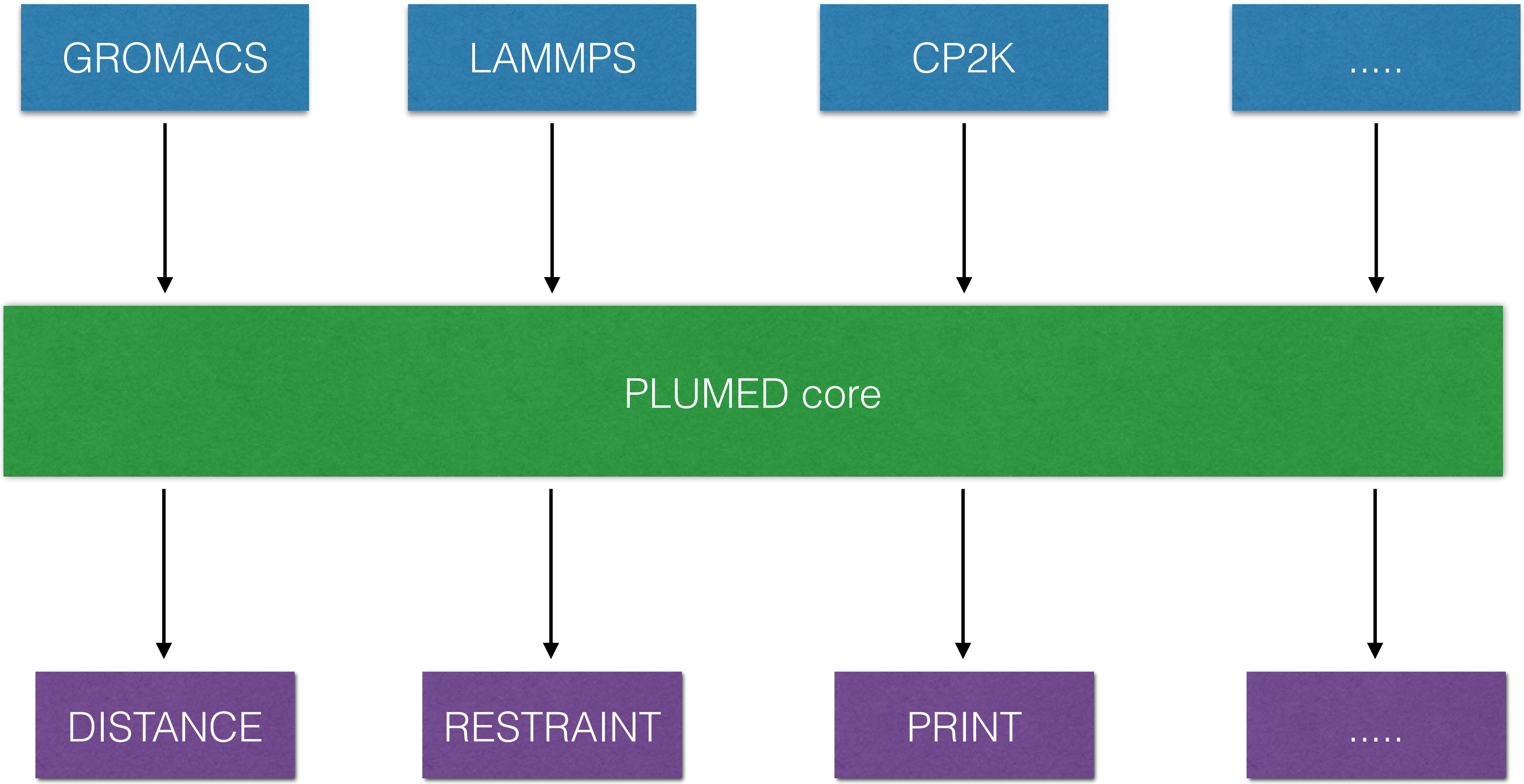
Reporting bugs and, when possible, solutions

Helping us improving the documentation

Programming new features - more in the next slides

* on your blog, on the SI of your papers, etc; also installation tricks are useful

Some details about PLUMED structure



GROMACS

LAMMPS

CP2K

.....

PLUMED core

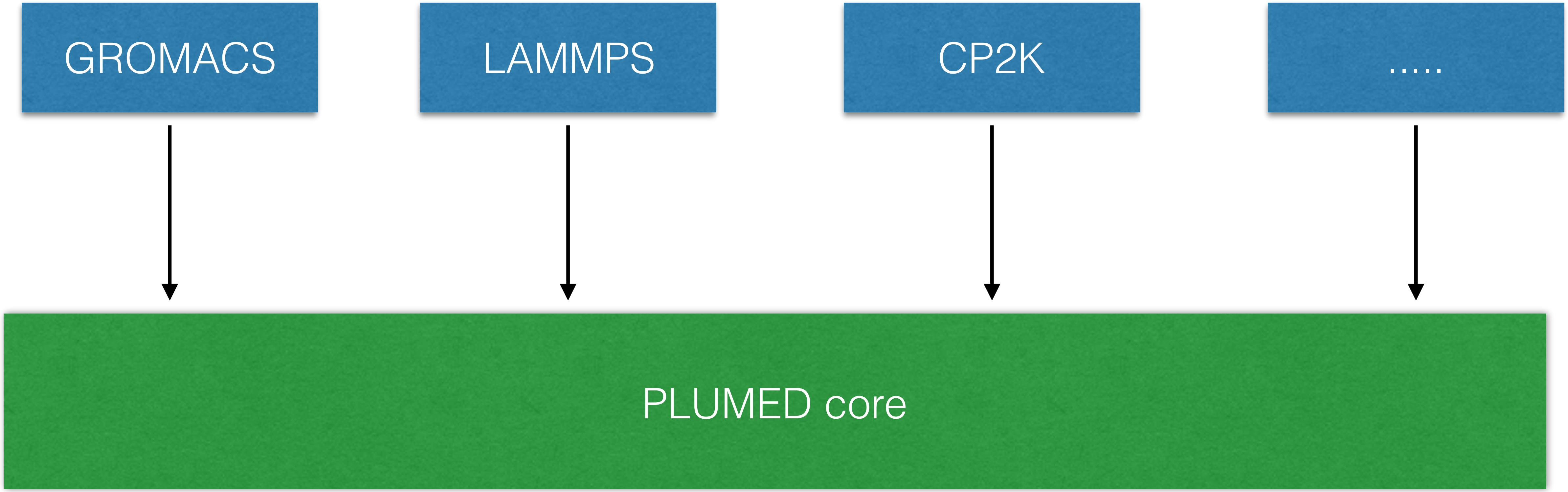
DISTANCE

RESTRAINT

PRINT

.....

Interfacing with MD codes



PLUMED is a library

C interface (some documentation + working examples)

Patches (e.g. GROMACS)
Some code implements calls to PLUMED (e.g. CP2K)

Contribute an MD interface

Easy (if you know well the MD code you want to modify)

PLUMED C interface is stable and backward compatible

No need to change PLUMED

No need to inform us*, just tell it to your users

* if you do it, we will advertise your code in the PLUMED manual

Examples

OPENMM interface: written by Peter Eastman

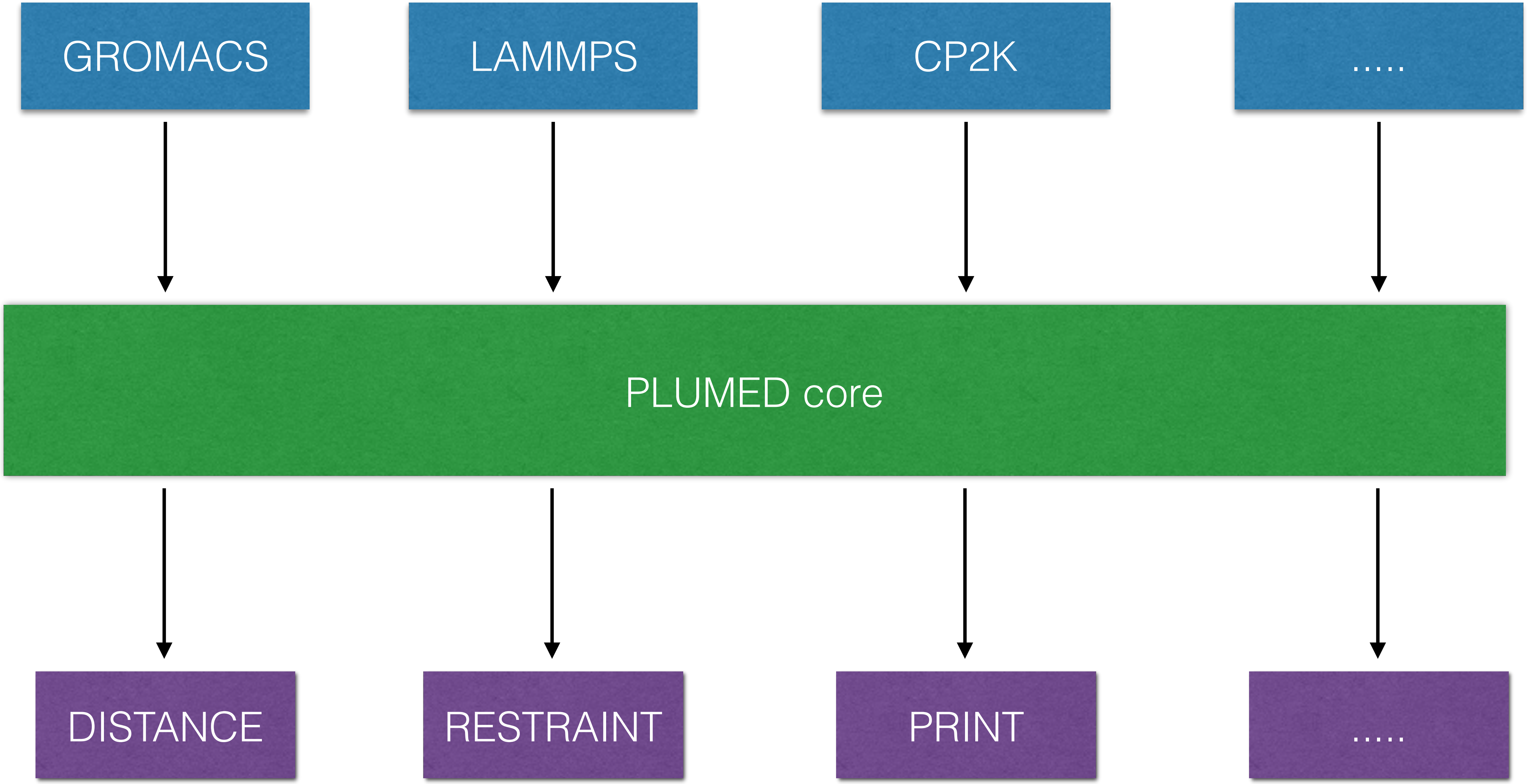
CP2K interface: written by Iain Bethune

PINY-MD interface: written by Ondrej Marsalek

ACEMD interface: written by ACEMD developers

...

Some details about PLUMED structure



Some details about PLUMED structure

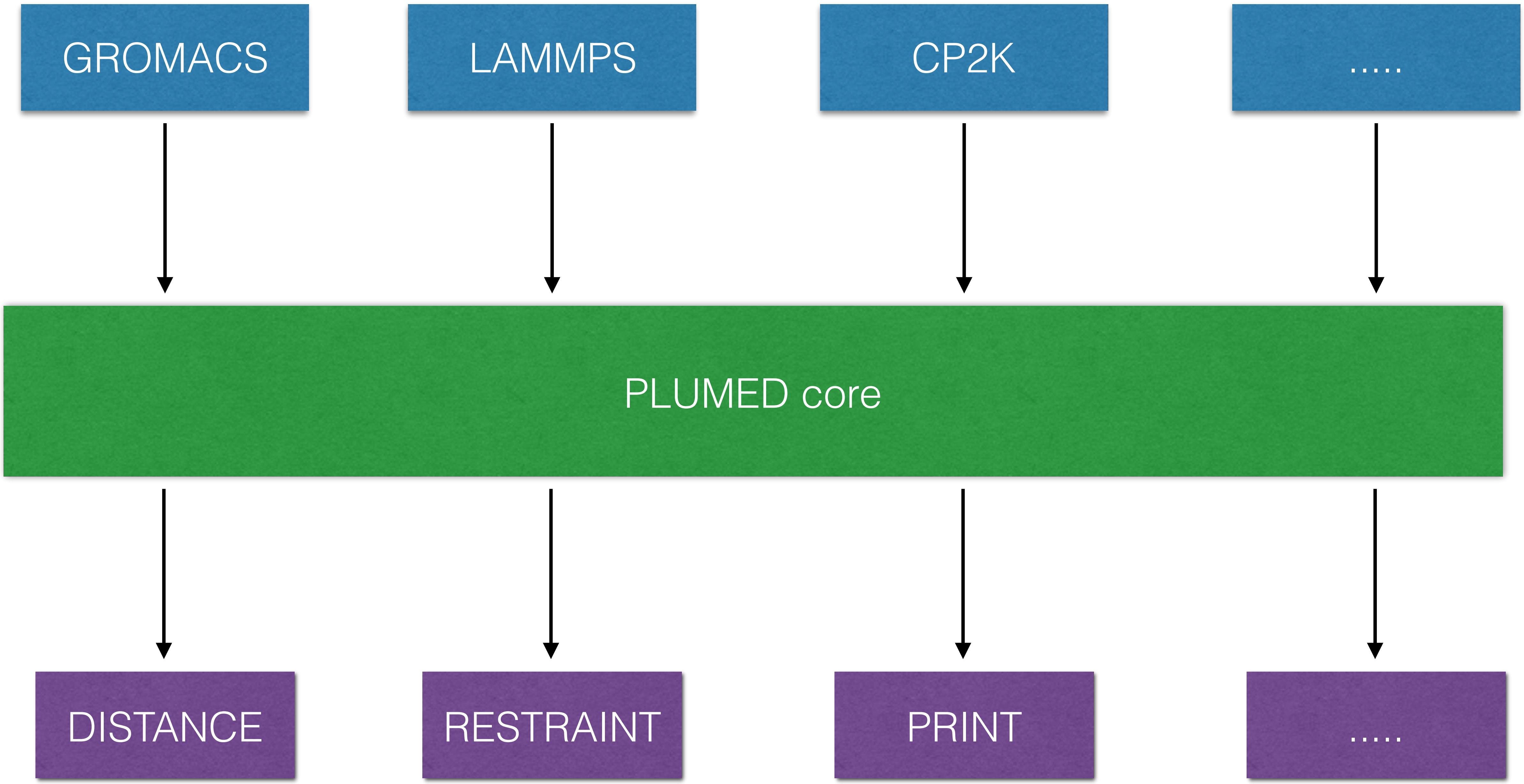
PLUMED core

This is the most complicated part.

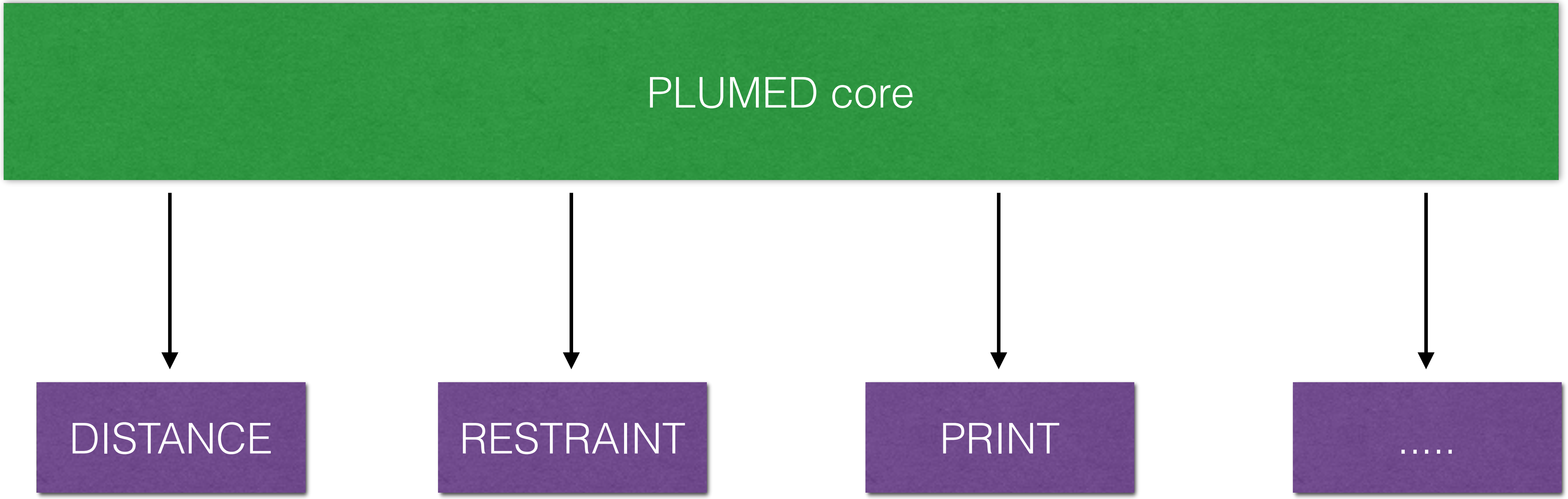
Mostly maintained by “core developers”
Occasional developers might report and even fix bugs

No discussion today!

Some details about PLUMED structure



Some details about PLUMED structure

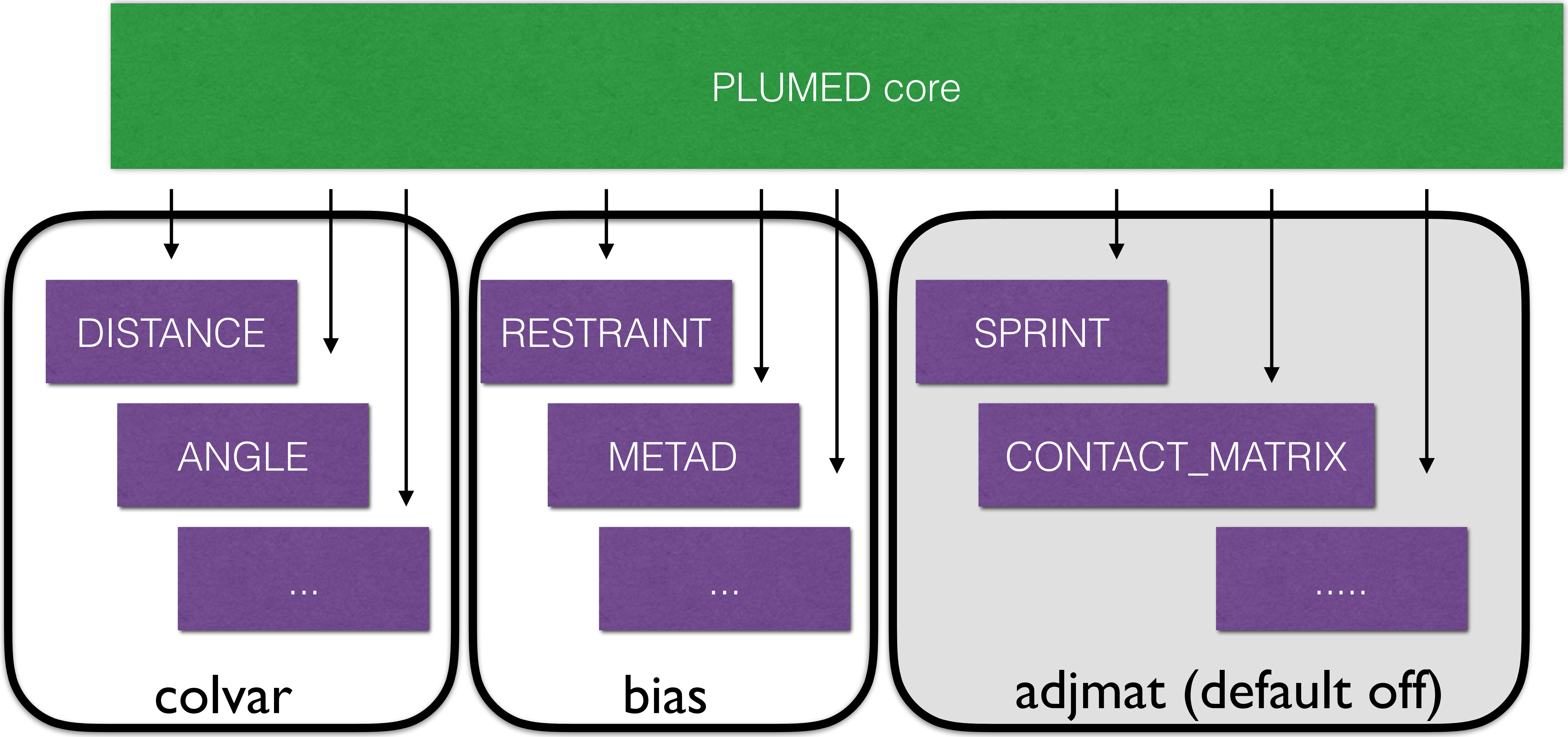


Each “action” (DISTANCE, etc) is coded as a C++ class

Implementation and user documentation are in the same file.

Actions can use the tools available from PLUMED core
(e.g. grids/pbc/etc)

PLUMED “modules”



Modules can have different license and authors
Can be on or off by default

Implement your stuff

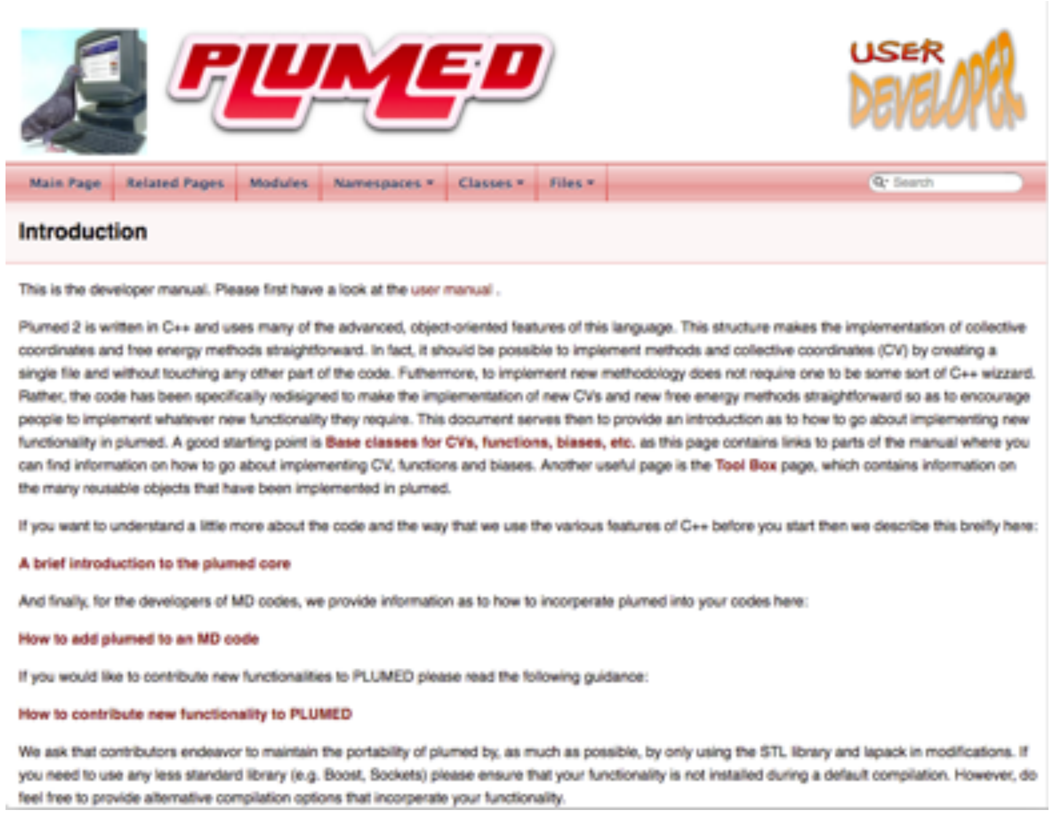
Check if what you need is already there

Write your C++ class.
Keep your C++ code documented

Add user documentation and examples

Run regtests to be sure you are not breaking other features

Add your regtests to be sure your stuff remains functional



Introduction

This is the developer manual. Please first have a look at the user manual .

Plumed 2 is written in C++ and uses many of the advanced, object-oriented features of this language. This structure makes the implementation of collective coordinates and free energy methods straightforward. In fact, it should be possible to implement methods and collective coordinates (CV) by creating a single file and without touching any other part of the code. Futhermore, to implement new methodology does not require one to be some sort of C++ wizzard. Rather, the code has been specifically redisigned to make the implementation of new CVs and new free energy methods straightforward so as to encourage people to implement whatever new functionality they require. This document serves then to provide an introduction as to how to go about implementing new functionality in plumed. A good starting point is **Base classes for CVs, functions, biases, etc.** as this page contains links to parts of the manual where you can find information on how to go about implementing CV, functions and biases. Another useful page is the **Tool Box** page, which contains information on the many reusable objects that have been implemented in plumed.

If you want to understand a little more about the code and the way that we use the various features of C++ before you start then we describe this breifly here:

A brief introduction to the plumed core

And finally, for the developers of MD codes, we provide information as to how to incorperate plumed into your codes here:

How to add plumed to an MD code

If you would like to contribute new functionalities to PLUMED please read the following guidance:

How to contribute new functionality to PLUMED

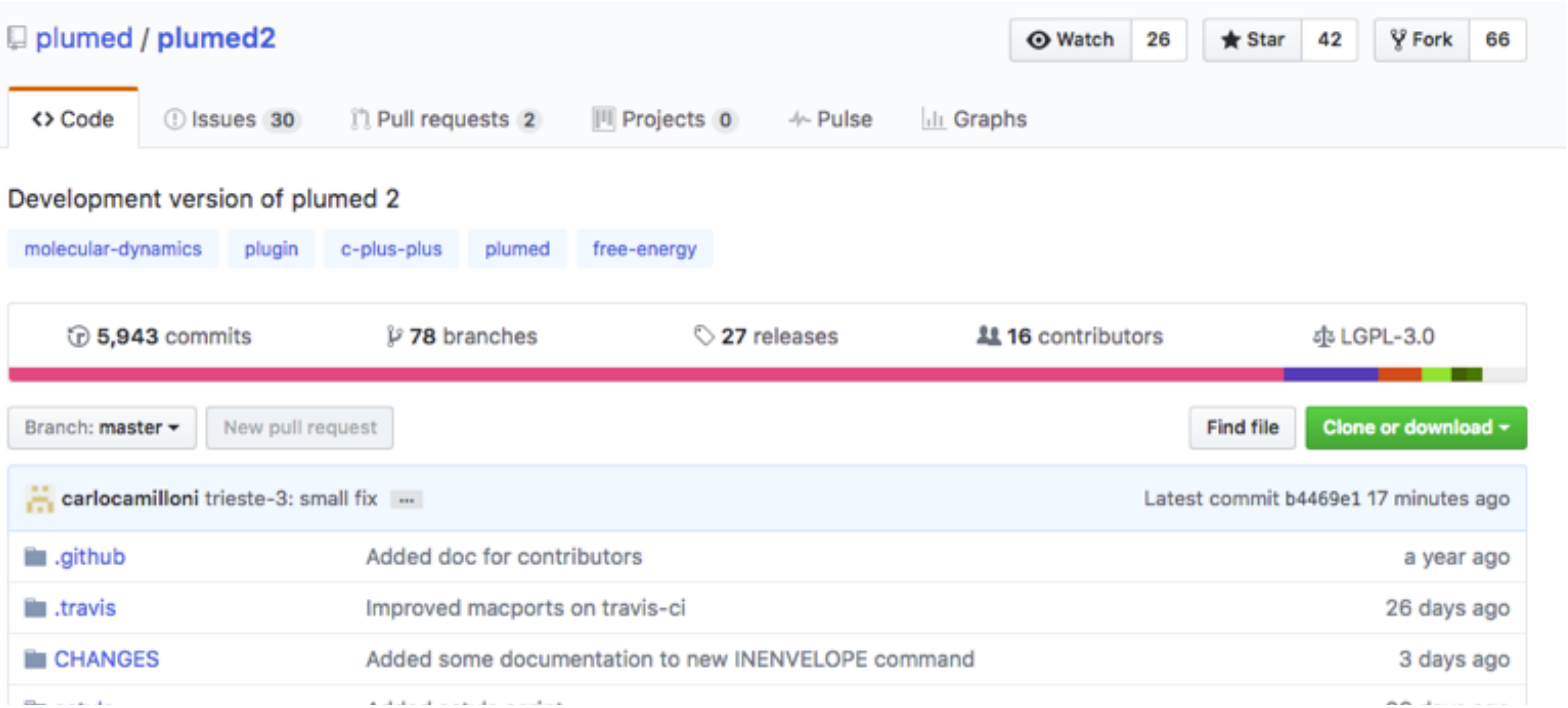
We ask that contributors endeavor to maintain the portability of plumed by, as much as possible, by only using the STL library and lapack in modifications. If you need to use any less standard library (e.g. Boost, Sockets) please ensure that your functionality is not installed during a default compilation. However, do feel free to provide alternative compilation options that incorperate your functionality.

Developer manual contains a lot of information

Share your code

PLUMED is L-GPL3, so you are allowed to redistribute your changes (and we are happy if you do it!)

Put your code on a github fork, so that it is visible and can be kept in sync. Or share it on your webpage if you like



plumed / plumed2

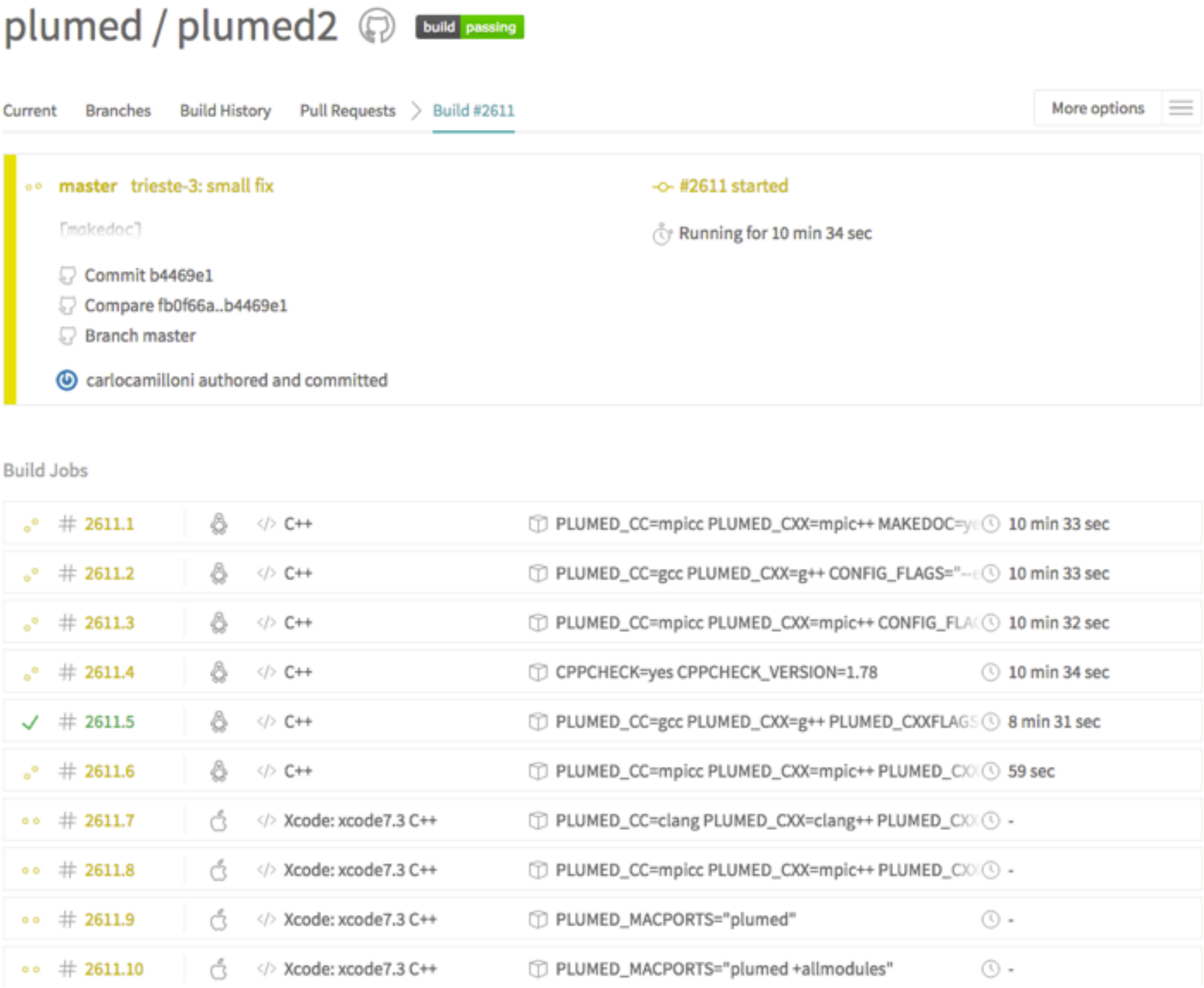
Development version of plumed 2

5,943 commits · 78 branches · 27 releases · 16 contributors · LGPL-3.0

carlocamilloni trieste-3: small fix … Latest commit b4469e1 17 minutes ago

.github	Added doc for contributors	a year ago
.travis	Improved macports on travis-ci	26 days ago
CHANGES	Added some documentation to new INENVELOPE command	3 days ago

Things to know: Travis-CI



plumed / plumed2 build passing

Current · Branches · Build History · Pull Requests · Build #2611 · More options

master trieste-3: small fix

[makedoc]

Commit b4469e1

Compare fb0f66a..b4469e1

Branch master

carlocamilloni authored and committed

#2611 started

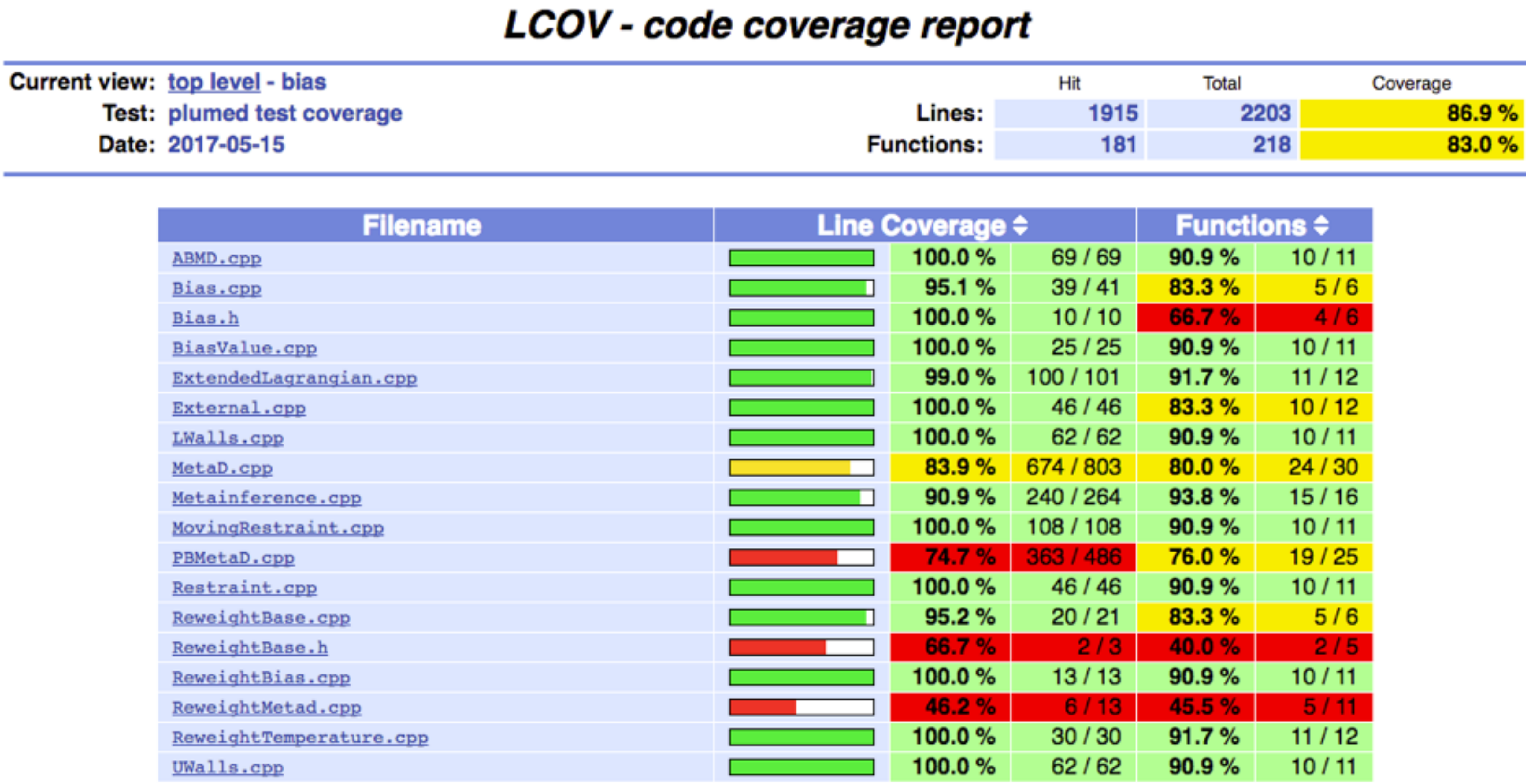
Running for 10 min 34 sec

Build Jobs

	# 2611.1	C++	PLUMED_CC=mpicc PLUMED_CXX=mpic++ MAKEDOC=ye	10 min 33 sec
	# 2611.2	C++	PLUMED_CC=gcc PLUMED_CXX=g++ CONFIG_FLAGS="--e	10 min 33 sec
	# 2611.3	C++	PLUMED_CC=mpicc PLUMED_CXX=mpic++ CONFIG_FLA	10 min 32 sec
	# 2611.4	C++	CPPCHECK=yes CPPCHECK_VERSION=1.78	10 min 34 sec
✓	# 2611.5	C++	PLUMED_CC=gcc PLUMED_CXX=g++ PLUMED_CXXFLAGS	8 min 31 sec
	# 2611.6	C++	PLUMED_CC=mpicc PLUMED_CXX=mpic++ PLUMED_CX	59 sec
	# 2611.7	Xcode: xcode7.3 C++	PLUMED_CC=clang PLUMED_CXX=clang++ PLUMED_CX	-
	# 2611.8	Xcode: xcode7.3 C++	PLUMED_CC=mpicc PLUMED_CXX=mpic++ PLUMED_CX	-
	# 2611.9	Xcode: xcode7.3 C++	PLUMED_MACPORTS="plumed"	-
	# 2611.10	Xcode: xcode7.3 C++	PLUMED_MACPORTS="plumed +allmodules"	-

https://travis-ci.org/plumed/plumed2/

Things to know: coverage scan



LCOV - code coverage report

			Hit	Total	Coverage
Current view:	top level - bias				
Test:	plumed test coverage	Lines:	1915	2203	86.9 %
Date:	2017-05-15	Functions:	181	218	83.0 %

Filename	Line Coverage ⬍			Functions ⬍	
ABMD.cpp	🟩	100.0 %	69 / 69	90.9 %	10 / 11
Bias.cpp	🟩	95.1 %	39 / 41	83.3 %	5 / 6
Bias.h	🟩	100.0 %	10 / 10	66.7 %	4 / 6
BiasValue.cpp	🟩	100.0 %	25 / 25	90.9 %	10 / 11
ExtendedLagrangian.cpp	🟩	99.0 %	100 / 101	91.7 %	11 / 12
External.cpp	🟩	100.0 %	46 / 46	83.3 %	10 / 12
LWalls.cpp	🟩	100.0 %	62 / 62	90.9 %	10 / 11
MetaD.cpp	🟨	83.9 %	674 / 803	80.0 %	24 / 30
Metainference.cpp	🟩	90.9 %	240 / 264	93.8 %	15 / 16
MovingRestraint.cpp	🟩	100.0 %	108 / 108	90.9 %	10 / 11
PBMetaD.cpp	🟥	74.7 %	363 / 486	76.0 %	19 / 25
Restraint.cpp	🟩	100.0 %	46 / 46	90.9 %	10 / 11
ReweightBase.cpp	🟩	95.2 %	20 / 21	83.3 %	5 / 6
ReweightBase.h	🟥	66.7 %	2 / 3	40.0 %	2 / 5
ReweightBias.cpp	🟩	100.0 %	13 / 13	90.9 %	10 / 11
ReweightMetad.cpp	🟥	46.2 %	6 / 13	45.5 %	5 / 11
ReweightTemperature.cpp	🟩	100.0 %	30 / 30	91.7 %	11 / 12
UWalls.cpp	🟩	100.0 %	62 / 62	90.9 %	10 / 11

Link can be found in developer manual

Contribute small changes

1. Check that the modified code still passes the regtests
2. Add regtests for the features you added
3. Send us your changes or, better, open a pull request on github

You should also transfer the copyright of the code to us.

We will be responsible for maintaining your changes

Examples

- VMD molfiles[*] plus small fixes[%], by Toni Giorgino
- Many small fixes[%] by Omar Valsson
- Improved LAMMPS interface[#] by Marco DeLaPierre
- Native contact (Q) switching function, by Jan Domanski
- ERMSD distance, by Sandro Bottaro
- Many new atom selections (e.g. @chi-1), by Alejandro Gil-Ley
- DIMER by Mark Nava
- R_POWER in COORDINATIONNUMBER by Andrew White
- ...

see also https://github.com/plumed/plumed2/graphs/contributors

[*] a separate module, with a different license

[%] fixes in the PLUMED core

[#] fixes in the LAMMPS patch

Contribute large changes

1. Put all your changes in an optional module
2. Check that the modified code still passes the regtests
3. Add regtests for the features you added
4. Send us your changes or, better, open a pull request on github

No need to transfer the copyright of the code to us.

You will be responsible for maintaining your code in the future

Notice that the code you contribute will not be enabled by default

Examples

- EDS module, contributed by Glen Hocky and Andrew White
- VES module, to be contributed by Omar Valsson
- ...

Summary

Contribute to documentation and share usage examples

Contribute MD interfaces:
- help us maintaining the existing ones
- make PLUMED work with more MD codes

Contribute collective variables/methods:
- small changes (copyright & responsibility transfer)
- larger changes (optional modules, you keep copyright)

Feedbacks from the audience?

Our policy for copyright

Module organization:

- should manual be organized by "modules" or by "type of methods"?*
- is it ok that contributed modules are off by defaults?

Any of you would like to contribute something?

*e.g. if a module contains a colvar and a bias, should they be grouped are scattered in colvar and bias sections?