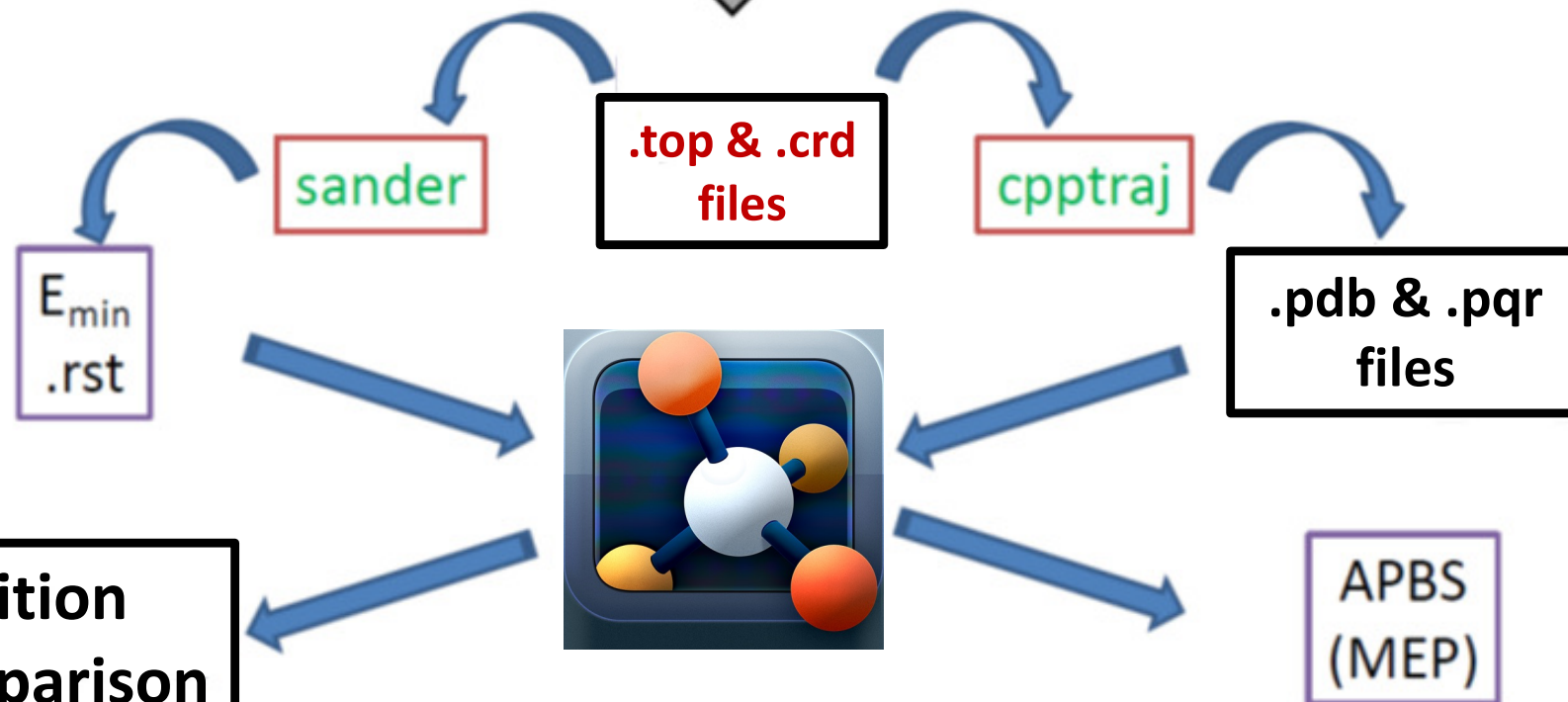
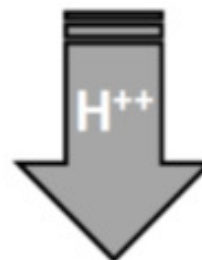


General workflow



Practical exercise

DHFR

Bibliographic source: Heaslet H, Harris M, Fahnoe K, Sarver R, Putz H, Chang J, Subramanyam C, Barreiro G, Miller JR.

Structural comparison of chromosomal and exogenous dihydrofolate reductase from *Staphylococcus aureus* in complex with the potent inhibitor **trimethoprim**.

Proteins 76(3):706-717 (2009). doi: 10.1002/prot.22383. PMID: 19280600.

PDB:

[2W9G](#) PyMOL session with superimposed protein structures: SaDHFR_PDB.pse

[2W9H](#)

[2W9S](#) Ligand: **TOP** (trimethoprim)

[2W9T](#) (apo)

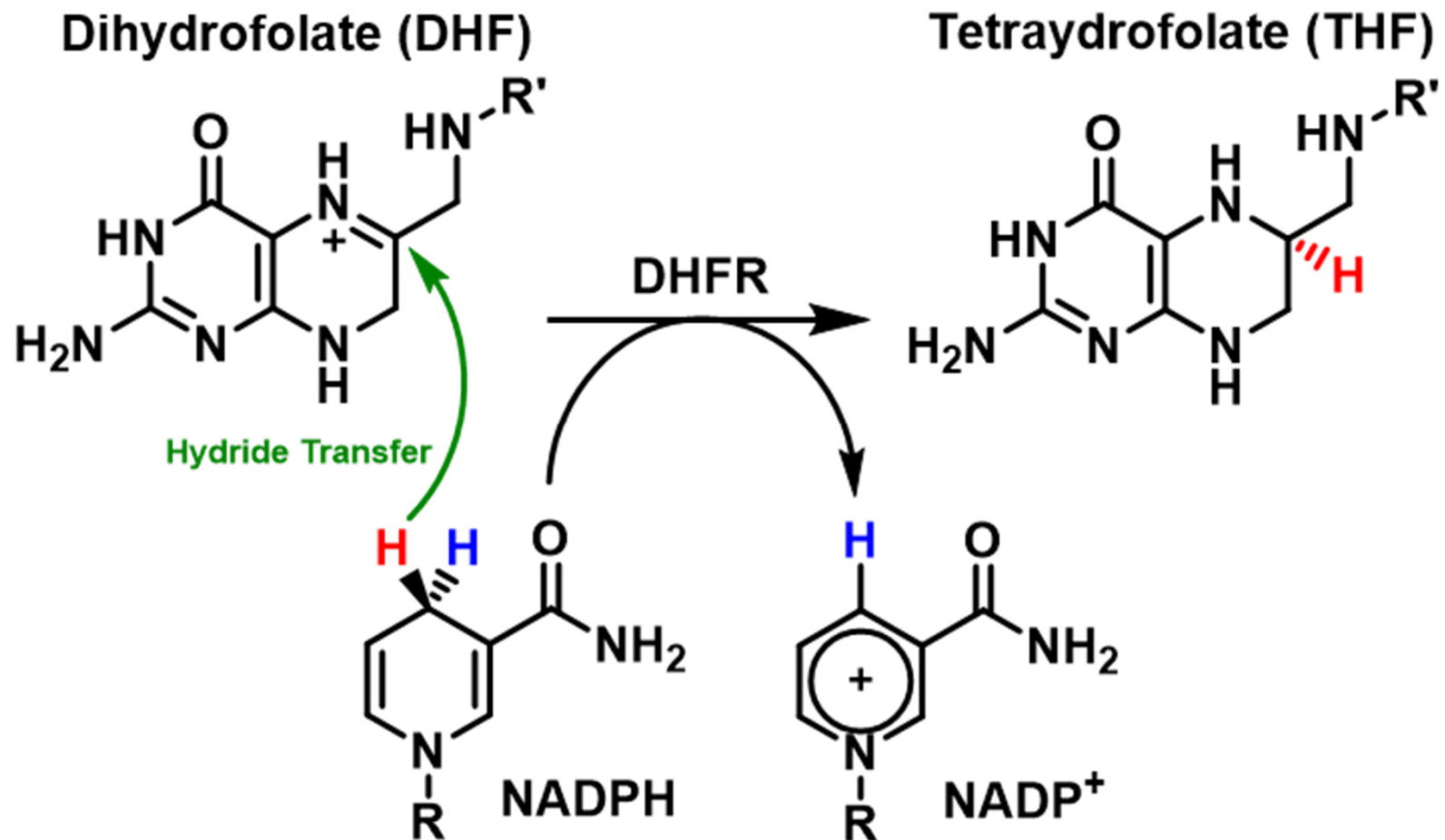
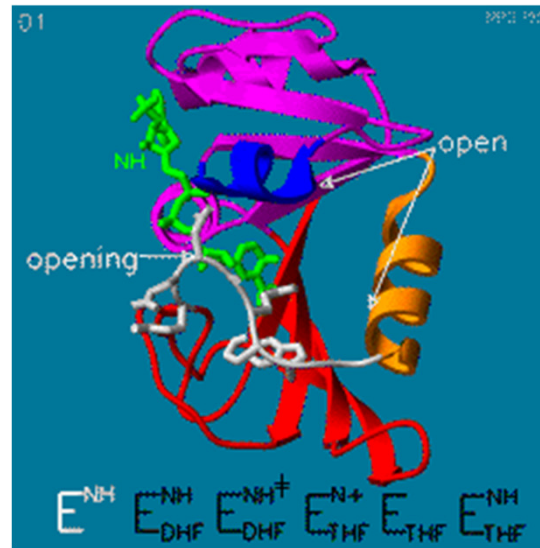


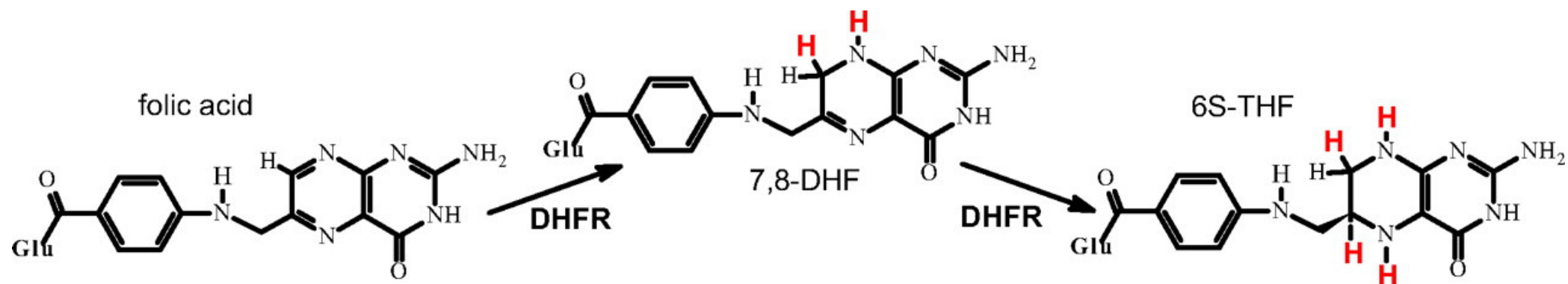
Secondary reference: Oefner C, Bandera M, Haldimann A, Laue H, Schulz H, Mukhija S, Parisi S, Weiss L, Lociuro S, Dale GE. Increased hydrophobic interactions of **iclaprim** with *Staphylococcus aureus* dihydrofolate reductase are responsible for the increase in affinity and antibacterial activity.

J Antimicrob Chemother. 63(4):687-698 (2009). doi: 10.1093/jac/dkp024. PMID: 19211577.

 **PDB:** [3FRD](#) Ligand: **DHF** (dihydrofolate)

Biochemical / Pharmacological background





```

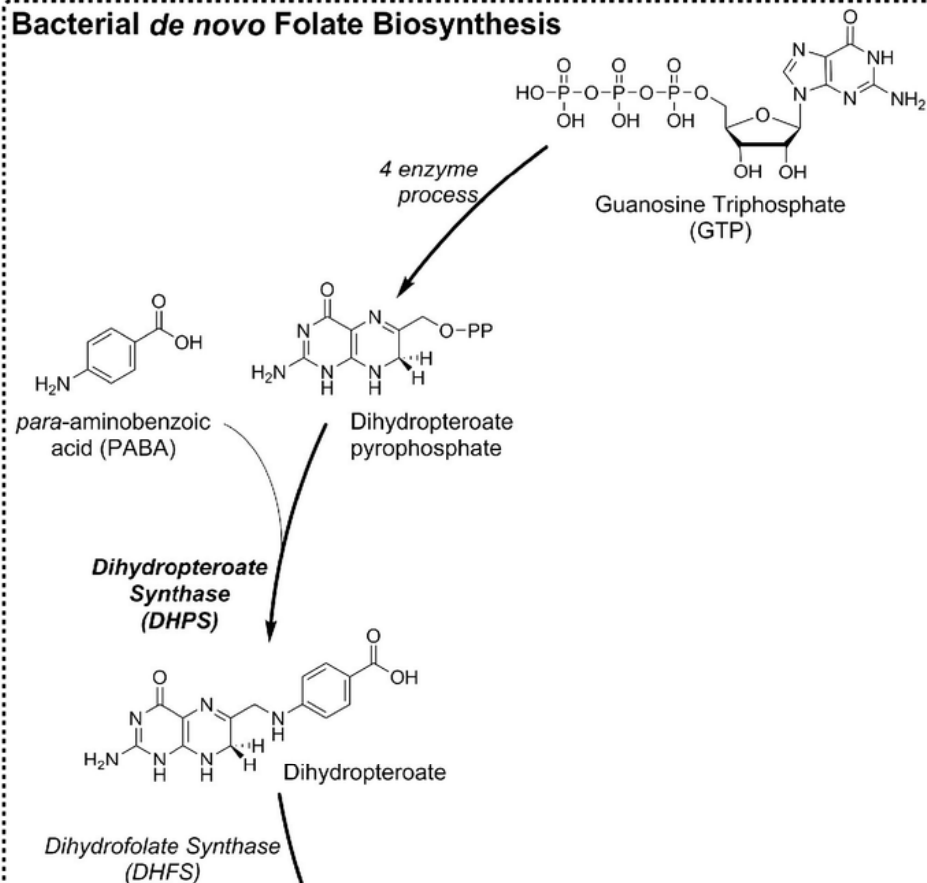
Ba DHFR  --MIVSFMVAMDENRVIGKDN NL PWRLP SELQYVKKTTMGH PL IMGRKNY  48
Sa DHFR  ---TL SILVAHDLQ RVIGFENQL PWHLPND LKHVKKLS TGH TL VMGRKTF  47
Ec DHFR  ---MI SL IAA LAV DRVIGMENAMPWNLP ADLAWFKRNT LDK PV IMGRHTW  47
Sp DHFR  MTKKI VAIWAQDEEGVIGKENRL PWHLP AE LQH FKETT LNHA I LMGRVTF  50
          ***                               **                               *

Ba DHFR  EAIG- RPLPGRRN IIVTRNEGYHVEGCEVAHSVEEVFELCKNEE-E I F I F  96
Sa DHFR  ESIG- KPLPNRRNVV LTS DT SFNVEGVDVI HSI EDIYQLPG ---HV F I F  92
Ec DHFR  ESIG- RPLPGRKN IILSSQPG-TDDRVTWVKSVD EAIACGDVP-E IMVI  94
Sp DHFR  DGMGRRL LPKRET LI LTRNPEEK IDGVATFQDVQSVLDWYQAQ EKNLY II  100
          **  *  *                               *                               *
          Y102                                     F96

Ba DHFR  GGAQI YDLFLPYVDKLYI TKIHHA FEGDTFFP-EMDMTNWKEVFVEKGLT  145
Sa DHFR  GGQTL FEEMI DKVDDMYI TVIEGKFRGDTFFP- PYTFEDWEVASSVEGKL  141
Ec DHFR  GGGRV YEQFLPKAQKLYL THIDAEVEGDTHFP-DYEPDDWESVFSEFHDA  143
Sp DHFR  GGKQI FQAFE PYLDEVIVTH IHARVEGDTYFPEELDLSLFETVSSKFYAK  150
          Y102                                     F96

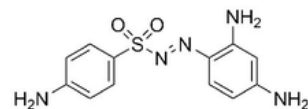
Ba DHFR  DEKN--PYTYYYHVYEKQQ  162
Sa DHFR  DEKNTIPHTFLHLIRK---  157
Ec DHFR  DAQN--SHSYCFEILERR-  159
Sp DHFR  DEKNPYDFTIQYRK RKEV-  168
  
```

Bacterial *de novo* Folate Biosynthesis

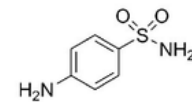


Clinically Used Antifolates

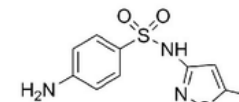
DHPS Inhibitors



Prontosil

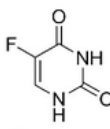


Sulfanilamide

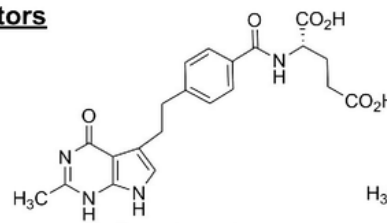


Sulfamethoxazole

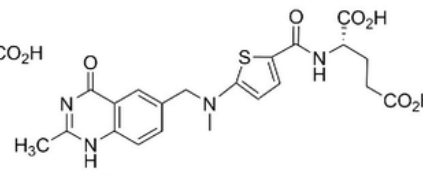
TS Inhibitors



Fluorouracil

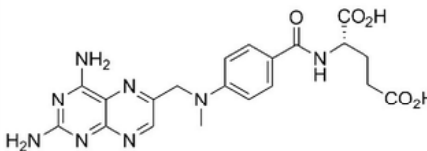


Pemetrexed

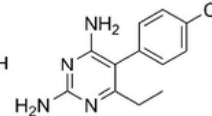


Raltitrexed

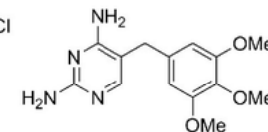
DHFR Inhibitors



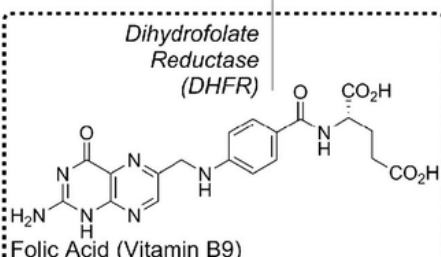
Methotrexate (MTX)



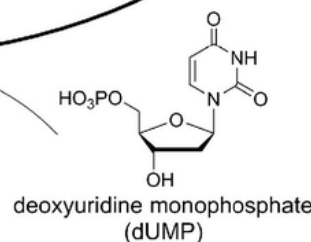
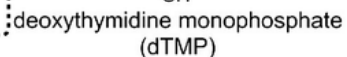
Pyrimethamine



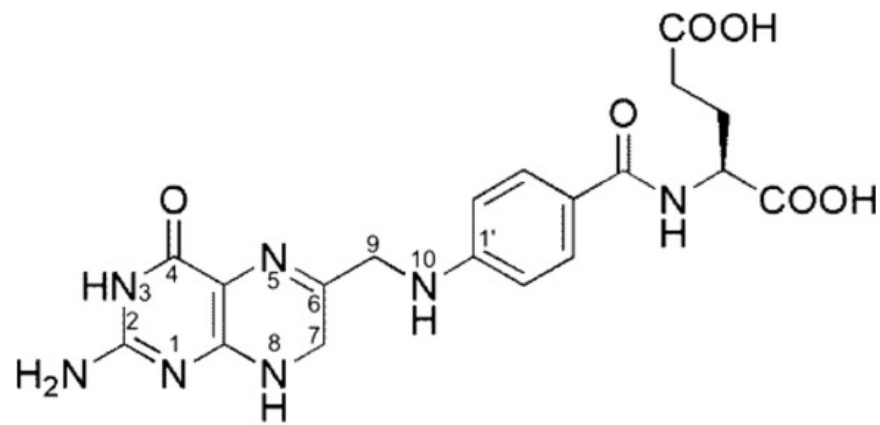
Trimethoprim (TMP)



Human Folate Acquisition

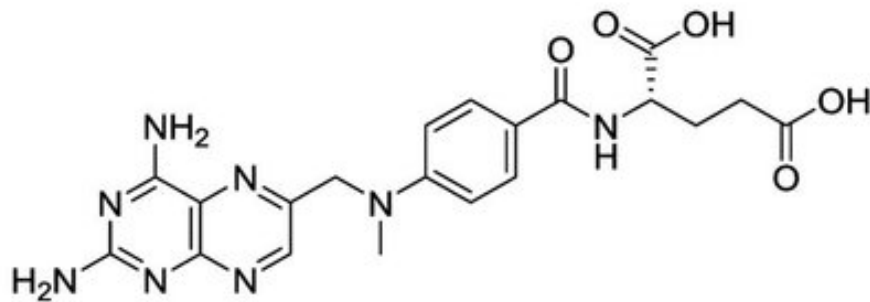


Biochemical / Pharmacological background

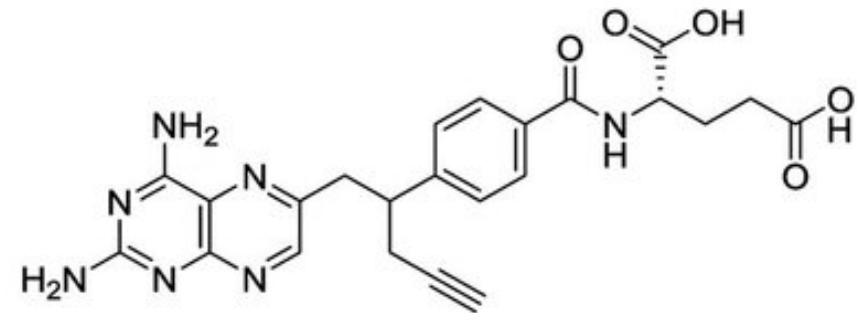


Dihydrofolate (DHF)

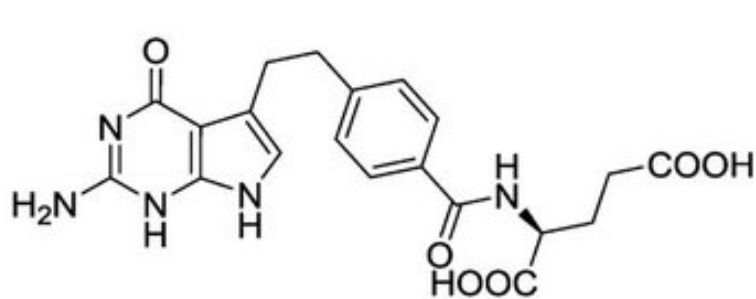
Classical antifolates



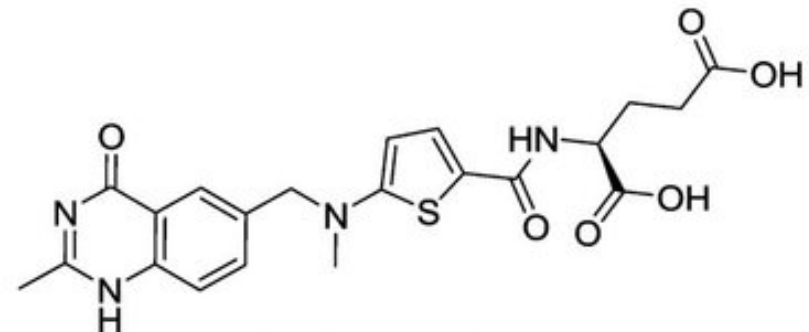
Methotrexate (MTX)



Pralatrexate (PDX)

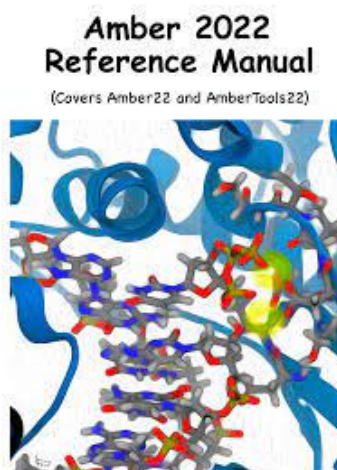


Pemetrexed (PMX)



Raltitrexed (RTX)

Practical exercise



DHFR

 AMBER (use *tLeap* to generate **.top** and **.crd** files for protein DHFR, both alone [apo form] and in complex with ligands)

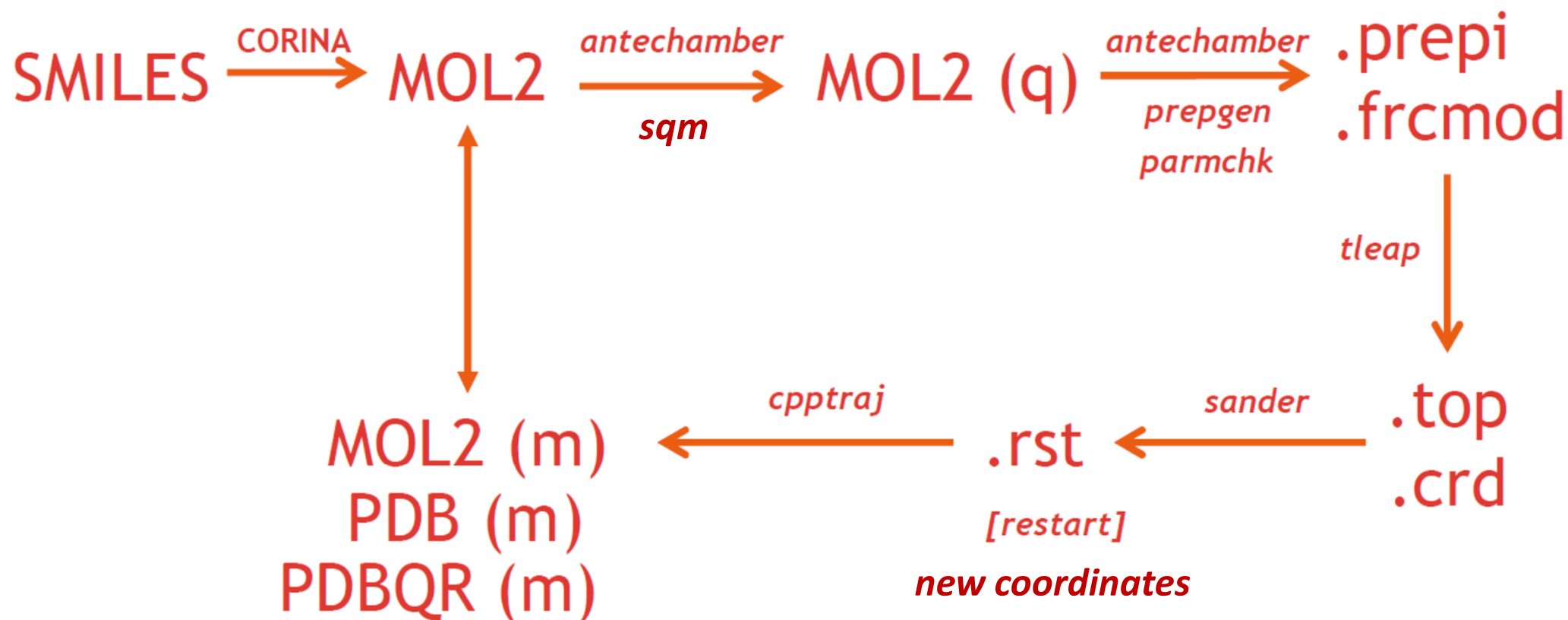
(1) *in vacuo*

(2) immersed in a rectangular box of TIP3P water

(3) immersed in a truncated octahedron of TIP3P water molecules

 antechamber (folder for ligand preparation and manipulation)

 complexes (folder for preparation and manipulation of L:P complexes)



MOL2 (q) \equiv MOL2 + atomic point charges

MOL2 (m) \equiv MOL2 (energy minimized)

script do_sqm.sh Ligand preparation: geometry optimization and derivation of atomic point charges.

Two consecutive steps:

(1) Geometry optimization by semiempirical QM program **sqm**

(2) Output file with optimized geometry + atomic point charges

```
#!/bin/bash
#ligand(s) with charge = 0
for n in TOP
do
antechamber -i $n.mol2 -fi mol2 -o tmp.$n.pdb -fo pdb -c bcc \
-ek 'maxcyc = 10 peptide_corr=1' -nc 0 -rn $n -rf $n -pf y
antechamber -i sqm.pdb -fi pdb -o $n.prp -fo prepi -c bcc \
-ek 'maxcyc = 0' -at gaff -nc 0 -rn $n -rf $n -pf y
mv -f sqm.pdb sqm.$n.pdb
Done
```

```
#ligand(s) with charge = -2
for n in DHF
do
antechamber -i $n.mol2 -fi mol2 -o tmp.$n.pdb -fo pdb -c bcc \
-ek 'maxcyc = 50 peptide_corr=1' -nc -2 -rn $n -rf $n -pf y
antechamber -i sqm.pdb -fi pdb -o $n.prp -fo prepi -c bcc \
-ek 'maxcyc = 0' -at gaff -nc -2 -rn $n -rf $n -pf y
mv -f sqm.pdb sqm.$n.pdb
done
```

antechamber -h

Usage: antechamber -i input file name
-fi input file format
-o output file name
-fo output file format
-c charge method
-cf charge file name
-nc net molecular charge (int)
-a additional file name
-fa additional file format
-ao additional file operation
 crd : only read in coordinate
 crg : only read in charge
 radius: only read in radius
 name : only read in atom name
 type : only read in atom type
 bond : only read in bond type
-m multiplicity (2S+1), default is 1
-rn residue name, overrides input file, default is MOL
-rf residue topology file name in prep input file,
 default is molecule.res
-ch check file name for gaussian, default is 'molecule'
-ek mopac or sqm keyword, inside a pair of quotes
-gk gaussian job keyword, inside a pair of quotes
-gm gaussian memory keyword, inside a pair of quotes, such as "%mem=1000MB"
-gn gaussian number of processors keyword, inside a pair of quotes, such as
 "%nproc=8"
-gv add keyword to generate gesp file (for Gaussian 09 only)
 1 : yes
 0 : no, the default
-ge gaussian esp file generated by iop(6/50=1), default is g09.gesp
-df am1-bcc precharge flag, 2 - use sqm(default); 0 - use mopac
-at atom type, can be gaff (the default), amber (for PARM94/99/99SB), bcc and sybyl



antechamber -h (cont.)

- du fix duplicate atom names: yes(y)[default] or no(n)
- bk 4-character component Id, for ccif
- an adjust atom names: yes(y) or no(n)
the default is 'y' for 'mol2' and 'ac' and 'n' for the other formats
- j atom type and bond type prediction index, default is 4
 - 0 : no assignment
 - 1 : atom type
 - 2 : full bond types
 - 3 : part bond types
 - 4 : atom and full bond type
 - 5 : atom and part bond type
- s status information: 0(brief), 1(default) or 2(verbose)
- eq equalizing atomic charge, default is 1 for '-c resp' and '-c bcc' and 0 for the other charge methods
 - 0 : no use
 - 1 : by atomic paths
 - 2 : by atomic paths and structural information, i.e. E/Z configurations
- pf remove intermediate files: yes(y) or no(n)[default]
- pl maximum path length to determine equivalence of atomic charges for resp and bcc, the smaller the value, the faster the algorithm, default is -1 (use full length),
set this parameter to 10 to 30 if your molecule is big (# atoms >= 100)
- i -o -fi and -fo must appear; others are optional



antechamber -L list the supported file formats and charge methods:

List of the File Formats

file format type	abbre.	index		file format type	abbre.	index
Antechamber	ac	1		Sybyl Mol2	mol2	2
PDB	pdb	3		Modified PDB	mpdb	4
AMBER PREP (int)	prepi	5		AMBER PREP (car)	prepc	6
Gaussian Z-Matrix	gzmat	7		Gaussian Cartesian	gcrt	8
Mopac Internal	mopint	9		Mopac Cartesian	mopcrt	10
Gaussian Output	gout	11		Mopac Output	mopout	12
Alchemy	alc	13		CSD	csd	14
MDL	mdl	15		Hyper	hin	16
AMBER Restart	rst	17		Jaguar Cartesian	jcrt	18
Jaguar Z-Matrix	jzmat	19		Jaguar Output	jout	20
Divcon Input	divcrt	21		Divcon Output	divout	22
SQM Input	sqmcrt	23		SQM Output	sqmout	24
Charmm	charmm	25		Gaussian ESP	gesp	26
Component cif	ccif	27				

AMBER restart file can only be read in as an additional file.

List of the Charge Methods

charge method	abbre.	index		charge method	abbre.	index
RESP	resp	1		AM1-BCC	bcc	2
CM1	cm1	3		CM2	cm2	4
ESP (Kollman)	esp	5		Mulliken	mul	6
Gasteiger	gas	7		Read in charge	rc	8
Write out charge	wc	9		Delete Charge	dc	10



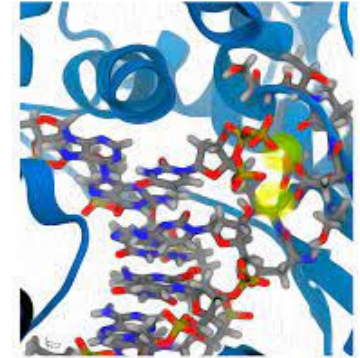
antechamber Some real-life examples



```
antechamber -i PM060184.mol2 -fi mol2 -o sqm.PM060184.mol2 -fo mol2 -c bcc -ek 'peptide_corr=1' -at sybyl -s 2 -nc 0 -pf y
antechamber -i sqm.PM060184.mol2 -fi mol2 -o PM060184.prep.ac -fo ac -c bcc -ek 'peptide_corr=1' -at amber -s 2 -nc 0 -pf y
antechamber -i PM060327.mol2 -fi mol2 -o sqm.PM060327.mol2 -fo mol2 -c bcc -ek 'peptide_corr=1' -at sybyl -s 2 -nc 0 -pf y
antechamber -i sqm.PM060327.mol2 -fi mol2 -o PM060327.prep.ac -fo ac -c bcc -ek 'peptide_corr=1' -at amber -s 2 -nc 0 -pf y
antechamber -i PM050489.mol2 -fi mol2 -o sqm.PM050489.mol2 -fo mol2 -c bcc -ek 'peptide_corr=1' -at sybyl -s 2 -nc 0 -pf y
antechamber -i sqm.PM050489.mol2 -fi mol2 -o PM050489.prep.ac -fo ac -c bcc -ek 'peptide_corr=1' -at amber -s 2 -nc 0 -pf y
antechamber -i FADHq-1_G09opt.pdb -fi pdb -o FDH.prp -fo prepi -j 3 -c bcc -ek 'maxcyc = 0' -at gaff2 -s 2 -nc -3 -pf y
antechamber -i 2F2_G09opt.pdb -fi pdb -o 2F2.prp -fo prepi -j 3 -c bcc -ek 'maxcyc = 0' -at amber -s 2 -nc 0 -pf y -rn 2F2 -rf 2F2
antechamber -i 2FAdo_G09opt.pdb -fi pdb -o 2FD.prp -fo prepi -j 3 -c bcc -ek 'maxcyc = 0' -at amber -s 2 -nc 0 -pf y -rn 2FD -rf 2FD
```


Input file for tleap: tleap.DHFR.vac.in

Amber 2022
Reference Manual
(Covers Amber22 and AmberTools22)



```
#!/bin/bash
```

```
# Use: tleap -f tleap.DHFR.vac.in
```

```
AMBERHOME=/software/amber22
```

```
addPath $AMBERHOME/dat/leap/cmd
```

```
addPath $AMBERHOME/dat/leap/lib
```

```
addPath $AMBERHOME/dat/leap/parm
```

```
source leaprc.protein.ff14SB
```

```
source leaprc.water.tip3p
```

```
loadoff nucleic12.lib
```

```
loadoff atomic_ions.lib
```

```
loadoff tip3pbox.off
```

```
loadamberparams frcmod.parmbsc1
```

```
# the next three lines only if it is the DHFR:DHF complex
```

```
#source leaprc.gaff2
```

```
#loadamberprep DHF.prp
```

```
#loadamberparams DHF.frcmod
```

```
unit = loadPDB ../PDB/DHFR.leap.pdb ### Prepared using H++
```


```
SaveAmberParm unit DHFR.vac.top DHFR.vac.crd
```

```
quit
```

Practical exercise

Solvating the system...



 **AMBER** (use of *tleap* to generate **.top** and **.crd** files for **protein DHFR** immersed in a rectangular box of TIP3P water)

```
#!/bin/bash
```

```
# Usage: tleap -f tleap.DHFR.Box.in
```

```
AMBERHOME=/software/amber22
```

```
addPath $AMBERHOME/dat/leap/cmd
```

```
addPath $AMBERHOME/dat/leap/lib
```

```
addPath $AMBERHOME/dat/leap/parm
```

```
source leaprc.protein.ff14SB
```

```
source leaprc.water.tip3p
```

```
loadoff nucleic12.lib
```

```
loadoff atomic_ions.lib
```

```
loadoff tip3pbox.off
```

```
loadamberparams frcmod.parmbsc1
```

```
# Now we load the refined structure (previously refined in vacuo)
```

```
unit = loadPDB ../DHFR.min.pdb
```

```
# Counterions are added to neutralize the system
```

```
addions unit Na+ 0
```

```
addions unit Cl- 0
```

```
SolvateBox unit TIP3PBOX 12.0 0.9
```


```
SaveAmberParm unit DHFR.Box.top DHFR.Box.crd
```

```
quit
```

Practical exercise

Solvating the system...

 **DHFR**

 **AMBER** (use of *tLeap* to generate **.top** and **.crd** files for **protein DHFR** immersed in a truncated octahedron of TIP3P water molecules)

```
#!/bin/bash
```

```
# Usage: tLeap -f tLeap.DHFR.Box.in
```

```
AMBERHOME=/software/amber22
```

```
addPath $AMBERHOME/dat/Leap/cmd
```

```
addPath $AMBERHOME/dat/Leap/lib
```

```
addPath $AMBERHOME/dat/Leap/parm
```

```
source leaprc.protein.ff14SB
```

```
source leaprc.water.tip3p
```

```
loadoff nucleic12.lib
```

```
loadoff atomic_ions.lib
```

```
loadoff tip3pbox.off
```

```
loadamberparams frcmod.parmbsc1
```

```
# Now we load the refined structure (previously refined in vacuo)
```

```
unit = loadPDB ../DHFR.min.pdb
```

```
# Counterions are added to neutralize the system
```

```
addions unit Na+ 0
```

```
addions unit Cl- 0
```

```
SolvateOct unit TIP3PBOX 12.0 0.9
```

```
SaveAmberParm unit DHFR.Oct.top DHFR.Oct.crd
```

```
quit
```

MOLECULAR DYNAMICS

General protocol:

```
GO.1.sh
1  #!/bin/bash -e
2
3  cd /data/users/farmamol/HIVp
4
5  export CUDA_VISIBLE_DEVICES=1
6  export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-6.5/lib64
7  export AMBEREXE=/usr/local/amber14/bin
8
9  $AMBEREXE/pmemd.cuda_SPFP -O -i min1.in -o $1.m1.out -p $1.top -r $1.m1.ncrst -c $1.crd -ref $1.crd
10 $AMBEREXE/pmemd.cuda_SPFP -O -i min2.in -o $1.m2.out -p $1.top -r $1.m2.ncrst -c $1.m1.ncrst -ref $1.m1.ncrst
11 $AMBEREXE/pmemd.cuda_SPFP -O -i min3.in -o $1.m3.out -p $1.top -r $1.m3.ncrst -c $1.m2.ncrst -ref $1.m2.ncrst
12 $AMBEREXE/pmemd.cuda_SPFP -O -i min.in -o $1.min.out -p $1.top -r $1.min.ncrst -c $1.m3.ncrst
13 #
```

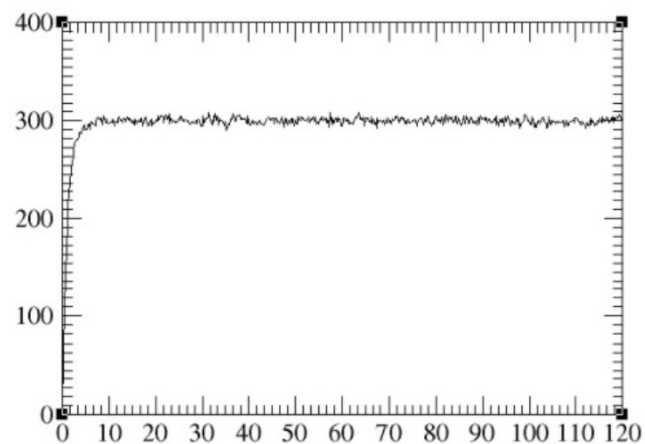
```
mn1.in
1  Minimizacion de todos los hidrogenos en GPU
2  &cntrl
3  ntx = 1, ntr = 1,
4  ntp = 500, igb=0,
5  ntf = 1, ntb = 1,
6  nsnb = 10, ntr = 1,
7  imin = 1, maxcyc = 500000, ncyc = 250000,
8  ntmin = 1, dx0 = 0.1, drms = 0.01,
9  ntp = 0, ntc = 1, cut=9., icutfm = 1,
10 restraint_wt=20., restraintmask='!@H-',
11 &end
12 &wt
13 type="END",
14 &end
```

```
mn2.in
1  Minimizacion de aguas y contralones + APL
2  &cntrl
3  ntx = 1, ntr = 1,
4  ntp = 500, igb=0,
5  ntf = 1, ntb = 1,
6  nsnb = 10, ntr = 1,
7  imin = 1, maxcyc = 50000, ncyc = 25000,
8  ntmin = 1, dx0 = 0.1, drms = 0.01,
9  ntp = 0, ntc = 1, cut=9., icutfm = 1,
10 restraint_wt=20., restraintmask=':1-199@CA',
11 &end
12 &wt
13 type="END",
14 &end
```

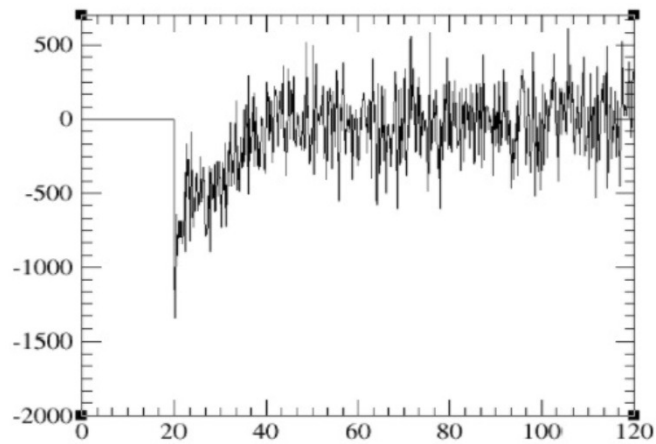
```
mn3.in
1  Minimizacion de todo excepto CA's de la proteina
2  &cntrl
3  ntx = 1, ntr = 1,
4  ntp = 500, igb=0,
5  ntf = 1, ntb = 1,
6  nsnb = 10,
7  imin = 1, maxcyc = 5000, ncyc = 1000,
8  ntmin = 1, dx0 = 0.1, drms = 0.001,
9  ntp = 0, ntc = 1, cut=9., icutfm = 1,
10 ntr = 1, restraint_wt=20., restraintmask=':1-199@CA',
11 &end
12 &wt
13 type="END",
14 &end
```

```
mn4.in
1  Energy minimization - no restraints
2  &cntrl
3  ntx = 1, ntr = 1,
4  ntp = 500, igb=0,
5  ntf = 1, ntb = 1,
6  nsnb = 10,
7  imin = 1, maxcyc = 25000, ncyc = 5000,
8  ntmin = 1, dx0 = 0.1, drms = 0.05,
9  ntp = 0, ntc = 1, cut=9., icutfm = 1,
10 ntr=0,
11 &end
12 &wt
13 type="END",
14 &end
```

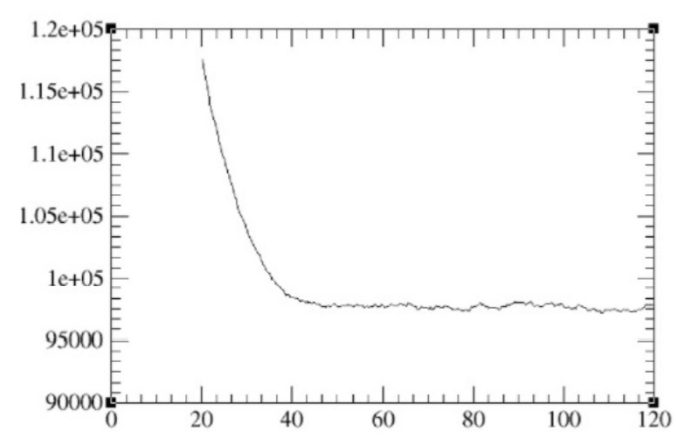
T_a vs t



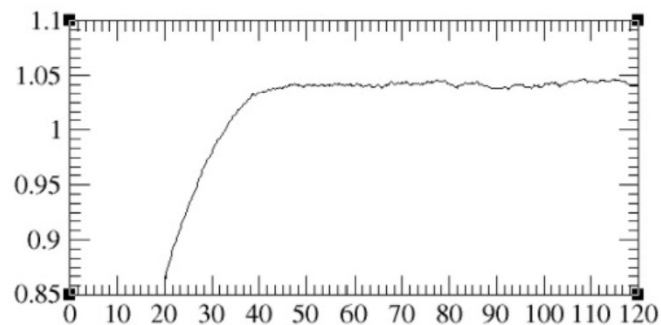
P vs t



V vs t



ρ vs t



E vs t

