pyPCAZIP: PCA-based trajectory file compression and analysis.

Introduction

Principal Component Analysis (PCA) forms the basis of a powerful range of simulation analysis methods, and also provides an approach to very efficient trajectory data compression. This document outlines the theory behind MD trajectory file compression using PCA and the tools developed to implement this that are distributed by the ExTASY project (www.extasy-project.org).

Background.

If a trajectory file consists of F snapshots of a system of N atoms, then the total number of (floating point) numbers in the trajectory file will be $3*N*F$.
If the trajectory is subjected to PCA, then it may be represented, exactly, by $3*N$ eigenvectors (each of size $3*N$), plus $3*N*F$ projections, which can regenerate the snapshots by operating on the time-averaged structure of the system ($3*N$ numbers).
But as:

$$3*N*F << (3*N)^2+3*N*F+3*N$$

this is a very inefficient way to represent the data. However, for a typical MD system, PCA will capture the vast majority of the motion of the system in a small number of eigenvectors, M ($M<<3*N$). So representation of the system, to an acceptable level of accuracy, becomes attractive in a PCA-based format if:

$$3*N*F > 3*N*M+M*F+3*N$$

For a typical biomolecular simulation - say 5000 snapshots of 700 atoms, we might expect the PCA analysis would be able to capture 90% of the variance in, perhaps, 35 eigenvectors. In which case:

$3*N*F$ = 10,500,000 numbers
$3*N*M+M*F+3*N$ = 250,600 numbers

i.e., we would have a compression to 2.3% of the original file size.

For a full discussion of this approach in practice, see: Mayer et al., *J. Chem. Theor. Comp.* 2006, **2**, 251-258.

pyPCAZIP is a Python-based reimplementation of the PCAZIP packages developed and distributed by the Laughton and Orozco groups. It offers essentially the same functionality and interface as those legacy codes, but is built on top of the new MDPlus python toolkit (Goni *et al.,* manuscript in preparation).**pyPcazip**

NAME

pyPcazip – compress MD trajectory files

SYNOPSIS

pyPcazip –i trajfile [trajfile2 ...] –o pczfile –t topfile [topfile2…] [-v] [-q quality] [-e eigenvectors] [-s selection [selection2 ...]] [-m maskfile] [-nofit] [-h]

OR:

pyPcazip –a albfile [albfile2 ...] –o pczfile –t topfile [topfile2 ...] [-v] [-q quality] [-e eigenvectors] [-s selection [selection2 ...]] [-m maskfile] [-nofit] [-h]

DESCRIPTION

pyPcazip compresses trajectory files using the PCA method, as described in: Mayer et al., *J. Chem. Theor. Comp.* 2006, **2**, 251-258.

REQUIRED

Either:

-i trajfile [trajfile2 ...]

One or more MD trajectory files to be compressed. Most common formats are supported. Compressed (.gz) files are readable too. The filename(s) can be modified with frame selections. Thus if *trajfile* is "traj1.mdcrd(23:50)" (and the quotes may be necessary to protect the string from the shell) then just the 23rd through 50th snapshots in traj1.x will be used. Other acceptable syntaxes are:

traj1.x(n)	selects just snapshot *n*
traj1.x(:e)	selects snapshots 1to *e*
traj1.x(b:)	selects snapshots from *b* to the end of the file
traj1.x(b:e:s)	selects every *s*-th snapshot from *b* to *e*
traj1.x(::s)	selects every *s*-th snapshot in the file

NB – pcazip uses zero-based indexing, so the first snapshot in the trajectory file is 0, not 1!

Or:

-a albfile [albfile2 ...]

An 'album' is simply a text file listing, one per line, the names of trajectory files to include in the analysis. Mixtures of different format files are permitted, The filenames can optionally be followed by a frame selection (see above).

Plus:

-o pczfile

The output compressed file. No suffix for this is enforced, though '.pcz' is encouraged, for consistency/transparency.

-t topfile [topfile2 ...]

One or more topology files that define the molecular systems present in the trajectory file(s) or album(s). If there is more than one trajectory file or album, then either there must be just one *topfile* specified that is common to all, or there must be one *topfile* given for every trajectory file/album.

OPTIONS

--c –centre

Places the atom(s) in *selection* in the centre of the periodic box. This may correct 'jumps' in the input trajectory caused by PBC effects, but the method is not perfect.

-e n_eigenvectors, -evecs n_eigenvectors

Fixed number of eigenvectors option. This overrides the selection of the number of eigenvectors to use in the compression based on a quality measure. Exactly n_*eigenvectors* (specified as an integer <= 3**natoms*) eigenvectors will be used.

-f file_version, --file_version file_version

Selects the PCZ file format that will be used (PCZ4, PCZ6, or PCZ7). The default HDF5-based (PCZ7) format is recommended as it is the most compact and portable, the other versions are included for backwards compatibility.

--fast

If specified, a fast, approximate, diagonalisation method is used for the PCA.

-h, --help

Prints a quick summary of options, and exits.

-l –lowmem

Reduces the memory footprint of the program, but may impact on performance.

-m maskfile [maskfile2..], –mask maskfile [maskfile2…]

Only include the subset of atoms present in *maskfile*, not every atom in *trajfile*, in the compressed file. *Maskfile* should be a PDB-format file. *Pcazip* will use the atom numbers in this – the second field in each ATOM record – as indices into the coordinate arrays in *trajfile*. So it is important that these are correct! If more than one trajectory or album file is specified, one or more corresponding mask files can be specified.

--nopca

Supresses the actual PCA process – useful as a quick check that the input files are coherent before a full calculation is performed.

-O --optimise

Optimises the set of trajectory files for parallel processing. Only useful in an MPI environment.

-p pdb_out, --pdb_out pdb_out

Outputs a PDB-format file that corresponds to the atoms selected from the full system by application of the *–mask* and *–selection* options. May serve as a useful topology file for other programs (e.g. *pczdump, pcaunzip* – see below)

-q quality, --quality quality

Quality option. By default, *pcazip* compresses files with enough eigenvectors to capture 90% of the variance in the data. This can be overridden by specifying *quality* (as number between 0 and 100, i.e. 0%-100% of variance captured).

-s selection [selection2 ...], –selection selection [selection2 ...]

selection is an MDAnalysis-style selection string (see http://mdanalysis.googlecode.com/git/package/doc/html/documentation_pages/selections.html) If there is more than one trajectory file or album, then either there must be just one *selection* specified that is common to all, or there must be one *selection* given for every trajectory file/album.

--trj_output trj_out

Writes the selected atoms in the selected trajectory frames to *trj_out*, which can be a .dcd or .xtc format file (as determined by the extension).

-v, --verbose

Verbose option. Various information messages are printed (to standard error) as the compression procedure progresses. '-vv' and '-vvv' give increasing levels of detail.

SEE ALSO

pcaunzip, pczdump, pczcomp, pczformat

DIAGNOSTICS/BUGS

Many and varied. *pyPcazip* will complain about missing input files, incompatible or unreadable trajectory files, etc. But the error catching is far from complete, so expect more-or-less obvious Python error messages…

EXAMPLES

pyPcazip –i trajfile.traj –o pczfile.pcz –t molecule.top

Compress *trajfile.traj*, whose molecular description is provided by *molecule.top*, producing the compressed file *pczfile.pcz*. Enough eigenvectors will be included to capture 90% of the variance.

pyPcazip –i trajfile.traj –o pczfile.pcz –t molecule.pdb –vv

As above, only produce progress messages as well, and uses a PDB format file for topology information.

pyPcazip –i trajfile.traj –o pczfile.pcz –t molecule.tpr –q 95

As above, only include enough eigenvectors to capture 95% of the variance.

pyPcazip -i trajfile.traj –o pczfile.pcz –t molecule.top –e 20

As above, only include exactly 20 eigenvalues in *pczfile.pcz*, whatever percentage of the variance this will capture.

pyPcazip –i trajfile.traj –o pczfile.pcz –t molecule.gro –q 95 –mask maskfile.pdb

Only include atoms present in *maskfile.pdb* in the compression procedure, and include enough eigenvectors to capture 95% of the variance. Be sure the atom numbers in *maskfile.pdb* are correct, and note that the '*-t*' argument specifies the full molecular system in the original *trajfile.traj*, not the selected subset.

pyPcazip –i trajfile.traj –o pczfile.pcz –t mol.gro –q 95 –s "protein and (not name H)"*

Only include protein heavy atoms in the compression procedure, and include enough eigenvectors to capture 95% of the variance. Note that the '*-t*' argument specifies the full molecular system in the original *trajfile.traj*, not the selected subset.

pyPcazip –i trajfile1.traj trajfile2.traj –o pczfile.pcz –t mol.gro –s "protein"

Process both trajectory files, each of which is compatible with the same topology file (*mol.gro*). Only include protein atoms in the compression procedure.

pyPcazip –i traj_dry.traj traj_hydrated.traj –o pczfile.pcz –t mol_dry.pdb mol_hydrated.pdb –s "protein"

Process both trajectory files, each of which has it's own associated topology file. Only include protein atoms in the compression procedure.

AUTHORS

Ardita Shkurti, Ramon Goni, Elena Breitmoser, Pau Andrio, Charlie Laughton 2014

py**Pcaunzip**

NAME

pyPcaunzip – uncompress MD trajectory files that have been compressed with *pyPcazip*

SYNOPSIS

pyPcaunzip –c pczfile –t topfile –o trajfile [-f fmt] [-v]

DESCRIPTION

Pcaunzip decompresses trajectory files that have been compressed using *pyPcazip*. Note that because the compression process is (adjustably) 'lossy', uncompressed files will not be identical to the original trajectory file.

REQUIRED

-c pczfile, --compressed pczfile

The compressed ('pcz format') MD trajectory file.

-t topfile, --topology topfile

The topology file that defines the molecular system. A range of formats for this are accepted – see *pyPcazip* documentation for details.

-o trajfile, --output trajfile

The uncompressed trajectory file. A range of output formats are available (Amber mdcrd, CHARMM dcd, GROMACS xtc, etc.). The required format is deduced from the filename extension but can be overridden by use of the optional *–format* argument (below)

OPTIONS

-format fmt

Selects the output trajectory format. Overrides what might have been deduced from the extension to *trajfile*. Options are "ncdf", "dcd", "xtc", "trr", and "trz".

SEE ALSO

pyPcazip, pyPczdump, pyPczcomp, pyPczformat

DIAGNOSTICS/BUGS

pyPcaunzip will complain about missing or wrong format input files, but the error catching is far from complete, so expect more-or-less obvious Python error messages…

EXAMPLES

pyPcaunzip –c pczfile.pcz –t molfile.top –o trajfile.ncdf

Uncompress *pczfile.pcz* to *trajfile.ncdf – an AMBER netcdf format file.*

pyPcaunzip –i pczfile.pcz –t molfile.pdb -o trajfile.xtc

Equivalent to the above, except a GROMACS xtc format file is produced.

AUTHORS

Ardita Shkurti, Ramon Goni, Elena Breitmoser, Pau Andrio, Charlie Laughton 2014

py**Pczdump**

NAME

pyPczdump – extract various data from a MD trajectory file compressed using pyPcazip

SYNOPSIS

pyPczdump option –i pczfile [option arguments]

DESCRIPTION

pyPczdump extracts various pieces of information from MD trajectory files compressed using pyPcazip.

REQUIRED

option

The type of analysis required. Options are one of: "--help", "--info", "--avg", "--evals", "--evec", "--proj", "--fluc", "--anim", "--rms", or "--coll". The additional requirements for each option are documented below.

-i pczfile, --input pczfile

The compressed ('pcz format') MD trajectory file to be analysed.

OPTIONS

-h, --*help*

Summarise usage options and syntax.

-n, --info

Print basic information about the data in *pczfile*. This includes the title in the original MD trajectory file, the numbers of atoms, frames and eigenvectors, and the quality (%age of variance captured).

--avg --pdb reffile –o avg.pdb

Output the time averaged structure, in PDB format, to *avg.pdb*. A suitable reference pdb file is also required as input, to define atom and residue names, etc.

-l, --evals

Print out the eigenvalues associated with each eigenvector present in *pczfile*, in order of decreasing magnitude.

-e iv, --evec iv

Print out the *iv*th eigenvector (indexed from zero). The most important (largest eigenvalue) eigenvector is the lowest-numbered.

-p iv, --proj iv

Print out the projections of the *iv*th eigenvector (1 value for each frame in the trajectory)

-f iv, --fluc iv

Print out the atomic fluctuations associated with the *iv*th eigenvector.

-m iv, --anim iv –-pdb reffile –o anim.pdb

Produce a multi-model PDB file animating the motion of the molecule along the *iv*th eigenvector/principal component. As with the –avg option, as reference PDB-format file must be supplied as input to define atom and residue names, etc.

-r iref --rms iref

Print out the rmsd of each frame in *pczfile* from the structure in snapshot *iref* (remember the 1st snapshot is iref=0). Note that depending on the quality setting used in the original compression, values calculated this way will not exactly match those obtained from calculations done on the original data, but for typical quality settings (90% or greater) the difference is small. If *iref* is -1, rmsds from the time-average structure will be output.

-c --coll

Prints out the collectivity metric, κ, for each eigenvector (see Brunschweiler et al, *J. Chem. Phys.*, 1995, **102**, 3396-3403). Modes that produce the most collective motion in the system will have high κ values, while modes that, for example, originate from large displacements of small substructures, will show low κ values.

SEE ALSO

pyPcazip, pyPcaunzip, pyPczcomp, pyPczformat

DIAGNOSTICS/BUGS

Many and varied. py*Pczdump* will complain about missing or wrong format input files, etc. But the error catching is far from complete, so expect more-or-less obvious Python error messages…

EXAMPLES

pyPczdump --info –i pczfile.pcz

Output (to the screen) basic information about the contents of *pczfile.pcz.*

pyPczdump --avg –i pczfile.pcz –o average.pdb --pdb reference.pdb

Extract the time average structure from *pczfile.pcz*, writing to *average.pdb*, using *reference.pdb* as a template.

pyPczdump --rms -1 –i pczfile.pcz

Calculate the rmsd between each frame in the MD trajectory and the time-averaged structure.

pyPczdump –rms 0 –i pczfile.pcz

Calculate the rmsd between each frame in the MD trajectory and the first frame.

pyPczdump --anim –i pczfile.pcz --evec 2 --pdb ref.pdb –o pc2_animation.pdb

Produce a multi-model pdb format file *pc2_animation.pdb* that animates the motion of the structure, about its time averaged position, along the **third** principal component (remember pyPcazip tools count from 0).

AUTHORS

Ardita Shkurti, Ramon Goni, Elena Breitmoser, Pau Andrio, Charlie Laughton 2014

py**Pczcomp**

NAME

pyPczcomp – compare two MD trajectories, compressed using pcazip

SYNOPSIS

pyPczcomp --input pczfile1 pczfile2 [--nvecs n_eigenvectors] [--quick] [-h]

DESCRIPTION

A very simple utility to do a basic comparison between two MD trajectory files that have been compressed using *pcazip*. The output contains the following information:

1. The RMSD between the two time-average structures contained in *pczfile1* and *pczfile2*.
2. The dot product matrix of the top *n* eigenvectors in *pczfile1* (x-axis), with those in *pczfile2* (y-axis). By default *n* is 10, but this can be overridden.
3. The subspace overlap.
4. Optionally, a range of metrics related to Mahalanobis distances can be calculated (for a discussion of Mahalanobis distances, see http://en.wikipedia.org/wiki/Mahalanobis_distance).

REQUIRED

-i *pczfile1 pczfile2, --input pczfile1 pczfile2*

The two compressed ('pcz format') MD trajectory files to be compared.

OPTIONS

-n n_eigenvectors

Use *eigenvectors* (specified as an integer less than or equal to the number of eigenvectors stored in *pczfile1* or *pczfile2*, whichever is the smaller) eigenvectors in the calculation of Mahalanobis distances, the dot product matrix, and the subspace overlap. In the absence of this option, ten eigenvectors (or the maximum possible, if less) are used.

--quick

Skip the calculation of the Mahalanobis metrics. This can be time-consuming for large datasets.

SEE ALSO

pyPcazip, pyPcaunzip, pyPczdump, pyPczformat

DIAGNOSTICS/BUGS

pyPczcomp will check that the two .pcz files have the same number of atoms per snapshot, but otherwise no test is made as to whether the comparison is going to be meaningful or not. py*Pczcomp* will complain about missing or wrong format input files, but the error catching is far from complete, so expect more-or-less obvious Python error messages.

EXAMPLES

py*Pczcomp –i pczfile1.pcz pczfile2.pcz --nvecs 6 --quick*

Compare *pczfile1.pcz* with *pczfile2.pcz*, using only six eigenvectors from each. Skip the time-consuming Mahalanobis distance based analyses.

AUTHORS

Ardita Shkurti, Ramon Goni, Elena Breitmoser, Pau Andrio, Charlie Laughton 2014

The format and contents of the files produced by py*Pcazip:*

DESCRIPTION

There are three pcazip file formats:

a) PCZ4: A platform-dependent 'pure' binary stream format,
b) PCZ6: A more highly compressed version. For a very small loss of precision compression can be improved by a factor of up to 2 compared to PCZ4. Still platform-dependent
c) PCZ7: An HDF5-based version of PCZ6 – slightly less compressed, but platform independent.

The PCZ4 file format

Files in PCZ4 format are pure binary, (platform-endian) sequential access files. They begin with a header as follows:

a) 4 byte string: the format identifier ('PCZ4')
b) 80 byte string: the title, taken from the trajectory file, (80 characters)
c) Three 4 byte integers: the number of atoms in a snapshot, snapshots (frames) in the trajectory, number of eigenvectors stored.
d) 4 byte float: the total of ALL the eigenvalues that were obtained when the original covariance matrix was diagonalised – not just the sum of those present in this file (this allows the quality of the file to be confirmed, even if it has become truncated).
e) Three 4-byte integers, reserved for future use
f) 4 byte integer: if >0, indicates that the file contains PDB-style information. If so, this is followed by one 16-byte block for each atom, each of which contains:
    i) 4 byte integer: the atom number
    ii) 4 byte string: the atom name
    iii) 4 byte integer: the residue number
    iv) 3 byte string: the residue name
    v) 1 byte character: the chain identifier

g) The time-average structure. Each of the 3*(no.of atoms) coordinates as a 4 byte float

h) The rest of the file consists of blocks, one per eigenvector stored. Each block contains the following data as 4 byte floats:

    i) The coefficients of the eigenvector (3*no. of atoms)
    ii) The eigenvalue
    iii) The projections of that eigenvector (one value for each snapshot)

The PCZ6 file format

This is exactly the same as PCZ4 format, except for the eigenvector blocks. Instead of storing each projection value p as a 4-byte float, we store a 2-byte integer, ip, from which the projection can be obtained as:

$$p = p\_mid + ip * p\_inc$$

where p_mid is the centre of the range of p values (i.e., (p_min + p_max)/2) and p_inc is (p_max - p_min)/65534). Thus:

h) The rest of the file consists of blocks, one per eigenvector stored. Each block contains the following data:

i) The coefficients of the eigenvector (3*no. of atoms) (4-byte floats)
ii) The eigenvalue (4 byte float)
iii) The centre of the range for that eigenvector (*p_mid*) (4-byte float)
iv) The projection increment (*p_inc*) (4-byte float)
v) The projections of that eigenvector (one value for each snapshot) (2-byte signed integer)

SEE ALSO

pyPcazip, pyPcaunzip, pyPczdump, pyPczcomp

AUTHORS

Ardita Shkurti, Ramon Goni, Elena Breitmoser, Pau Andrio, Charlie Laughton 2014