Robust Perron Cluster Analysis for coarse-graining of non-reversible stochastic matrices with complex eigenvalues

Anna-Simone Frank\textsuperscript{a,}\textsuperscript{*}, Alexander Sikorski\textsuperscript{b} and Susanna Röblitz\textsuperscript{a}

\textsuperscript{a}Computational Biology Unit, University of Bergen, Thormøhlensgate 55, Bergen, 5003, Norway
\textsuperscript{b}Zuse Institute Berlin, Takustrasse 7, Berlin, 14195, Germany

ARTICLE INFO

Keywords:
Complex eigenvalue problem
Spectral cluster analysis
Invariant sub-space condition
Matlab programming language
Computational efficiency

ABSTRACT

The Robust Perron Cluster Analysis (PCCA+) has become a popular algorithm for coarse-graining transition matrices of nearly decomposable Markov chains with transition states. Though originally developed for reversible Markov chains, it has been shown previously that PCCA+ can also be applied to cluster non-reversible Markov chains. However, the algorithm was implemented by assuming the dominant (target) eigenvalues to be real numbers. Therefore, the generalized Robust Perron Cluster Analysis (G-PCCA+) has recently been developed. G-PCCA+ is based on real Schur vectors instead of eigenvectors and can therefore be used to also coarse-grain transition matrices with complex eigenvalues. In its current implementation, however, G-PCCA+ is computationally expensive, which limits its applicability to large matrix problems.

In this paper, we demonstrate that PCCA+ works in fact on any dominant invariant subspace of a nearly decomposable transition matrix, including both Schur vectors and eigenvectors. In particular, by separating the real and imaginary parts of complex eigenvectors, PCCA+ also works for transition matrices that have complex eigenvalues, including matrices with a circular transition pattern. We show that this separation maintains the invariant subspace, and that our version of the PCCA+ algorithm results in the same coarse-grained transition matrices as G-PCCA+, whereby PCCA+ is consistently faster in runtime than G-PCCA+. The analysis is performed in the Matlab programming language and codes are provided.

1. Introduction

In order to understand cellular biological processes (e.g., gene regulatory networks [1, 2], biomolecular dynamics [3, 4, 5]), we need to interpret their dynamics by identifying structural changes on a network scale. This can be challenging given that biological processes occur on several different time-scales [6]. The available data, however, often represent only the fast time-scale [7, 6]. The data describing such fast processes can originate from, for example, mathematical models (e.g., the chemical master equation [2]), biomolecular simulations [8], or laboratory experiments [5]. The data are usually characterized by a huge number of system states, \(N\), and stochastic state transitions with short kinetic distances (i.e., on fast time-scales) [9]. The underlying stochastic process can be described mathematically as a discrete-time Markov chain with a short lag-time and short-lived states (also known as microstates) [10].

The system states and transition rates can be summarized into a transition matrix \(P\), which allows to calculate the specific probability that the process, in its next step, switches from one particular state to another [11]. Upon specific system conditions and assumptions (e.g., thermodynamic equilibrium), and by analyzing the eigendecomposition of the matrix \(P\), we can furthermore obtain additional information about the system dynamics [12]. Such information includes, for example, the identification of metastable sets (also known as macrostates). Macrostates represent long-lived system states on the slow time-scale (i.e., with long kinetic distance) and are therefore defined as sub-sets of the state space, in which the system spends a long time before transitioning to the next macrostate [13]. By reducing the system from (possibly infinitely) many microstates to a few macrostates, it is easier to analyze the global process structure and to interpret and understand the system dynamics and the general transition behavior of the Markov chain on long times scales [10].

By using spectral clustering methods, one can identify the macrostates through the eigen-decomposition of the matrix \(P\) [13]. An eminent example of such a method is the robust Perron Cluster Analysis (PCCA+), which groups

*Corresponding author
\begin{itemize}
  \item anna-simone.frank@uib.no (A. Frank);
  \item sikorski@zib.de (A. Sikorski);
  \item susanna.roblitz@uib.no (S. Röblitz)
\end{itemize}

ORCID(i)s: 0000-0002-3728-3476 (A. Frank);
0000-0001-9051-650X (A. Sikorski);
0000-0002-2735-0030 (S. Röblitz)
Spectral clustering of Markov chains with complex eigenvalues

microstates into macrostates such that transitions between macrostates are slow, and transitions within them are fast [10]. The result is a coarse-grained system representation, also known as Markov State Model (MSM), that describes the system behavior on a long time-scale [6, 12, 8]. A MSM consists of a finite, discrete state space of size \( N \) and a coarse-grained transition probability matrix, \( P_c \), which contains the conditional probabilities for transitions between the \( n \times n \) metastable sub-sets (macrostates) of the system [6].

The PCCA+ method, as described in [10], works for both reversible and non-reversible Markov chains with real dominant eigenvalues, but it does not consider cases with complex eigenvalues. Complex eigenvalues, however, are common for transition matrices of non-reversible Markov chains, and characteristic for circular chains. Therefore, in 2018, the PCCA+ method was extended to the Generalized PCCA+ (G-PCCA+) method [13], which employs a real Schur-decomposition instead of an eigen-decomposition. G-PCCA+ includes two sub-routines, one for the calculation of the full Schur-decomposition and one for sorting the Schur-vectors [13]. As reported in [14], these two sub-routines are computationally ineffective for large eigenvalue problems in Matlab.

This caveat of G-PCCA+ has encouraged us to introduce a faster but equally accurate algorithm for coarse-graining large stochastic matrices with complex eigenvalues. The method we present in this article is the same as the original PCCA+ [10], with the exception of specific adjustments that allow an efficient application to complex-valued eigenvalue problems. To highlight these modifications and to distinguish the method from the original PCCA+ and the G-PCCA+ algorithms, we will refer to it as cPCCA+ throughout this article, where ‘c’ stands for ‘complex’. To demonstrate the efficiency of cPCCA+, we will compare its results and runtimes with those of G-PCCA+.

The article is organized in the following way. We start by reviewing the concepts of the original PCCA+ algorithm and the recently developed G-PCCA+ algorithm, before we introduce the specific modifications of cPCCA+. For cPCCA+, we show theoretically and illustrate with examples that the invariant sub-space condition holds for transition matrices with complex eigenvalues, including circular Markov chains. We compare the runtimes of cPCCA+ and G-PCCA+ for increasing state-space size for two cases, a circular transition matrix and the generator of a stochastic gene-regulatory network. Finally, we discuss the results and present a conclusion. The appendix contains additional figures.

2. The PCCA+ algorithms

Decomposable Markov chains have transition matrices which, upon re-ordering of states according to the partitions, are block-diagonal. The PCCA+ algorithm as a spectral clustering method exploits the fact that the dominant eigenvectors of such matrices, i.e. the eigenvectors corresponding to the multiple eigenvalue one (Perron eigenvalue), are constant on the blocks. In other words, if there are \( n_c \) blocks and the matrix \( X \) contains the \( n_c \) dominant eigenvectors, \( X = [X_1, \ldots, X_{n_c}] \), then the rows of \( X \) that belong to states in the same block are identical. In fact, each row of \( X \) represents one of the \( n_c \) corners of an \((n_c - 1)\)-dimensional simplex (the reduction by one dimension is possible because \( X \) contains the constant vector). This simplex structure gets only slightly perturbed if the Markov chain is only nearly decomposable and if transition states occur [10]. Once this simplex structure is identified, the rows of \( X \), i.e. the states of the Markov chain, can be assigned to each of the \( n_c \) simplex corners/clusters in terms of fuzzy membership values according to their location within the simplex. For example, the centroid of a simplex would be assigned with membership value \( 1/n_c \) to each of the \( n_c \) clusters. In mathematical terms, finding this simplex structure and the membership values corresponds to finding a linear transformation \( A \)

\[
X = X A, \quad A \in \mathbb{R}^{n \times n_c} \text{ non-singular}
\]

s.th. \( \chi_j \geq 0 \quad \forall i \in \{1, \ldots, N\}, j \in \{1, \ldots, n_c\} \) (positivity)

\( \sum_{j=1}^{n_c} \chi_j(i) = 1, \quad \forall i \in \{1, \ldots, N\} \) (partition of unity),

where \( \chi_j \in \mathbb{R} \) are the membership vectors. They assign every state \( i \in \{1, \ldots, N\} \) in the state space to a cluster \( j \in \{1, \ldots, n_c\} \) with a certain grade of membership [10, 15].

In the following, we assume that the eigenvectors satisfy an orthonormality relation

\[
X^T D^2 X = I
\]

with identity matrix \( I \) and diagonal matrix \( D = \text{diag}(w) \), whereby \( w \) is some density vector, i.e. \( w > 0 \) and \( w^T 1 = 1 \).
**Remark 1.** In the original implementation of PCCA+, \( w \) was chosen to be the stationary density, i.e., \( w^T P = w^T \). Since the stationary density is very sensitive for nearly uncoupled Markov chains, its computation can be numerically unstable and lead to errors in the orthonormalization of the eigenvectors. The following results, however, are valid for any diagonal matrix \( D \), in particular also for \( w = (1/N, \ldots, 1/N)^T \).

As shown in [16], the problem of finding the right simplex and hence a suitable transformation matrix \( A \) does not have a unique solution. Hence, an optimization criterion needs to be defined to make the problem well-posed. The objective function in PCCA+ is defined as

\[
 n_c \rightarrow \text{trace}(D^{-2} \chi^T D^2 \chi) \rightarrow \min, \tag{3}
\]

whereby \( D^2 = \text{diag}(\chi^T w) \). This optimization results in membership vectors \( \chi \) that are as close to characteristic vectors (vectors containing only entries 0 and 1) as possible.

For a given number of clusters \( n_c \), the PCCA+ routine consists of two steps. First it computes an initial guess based on a heuristic, the inner simplex algorithm. It then proceeds to optimize the objective using an optimization method, leading to a locally optimal clustering. As optimization method, the user can currently choose between the Nelder-Mead simplex method, the Gauss-Newton method, and the Levenberg-Marquardt method. If the number of clusters is not known in advance, one can maximize the so-called crispness, \( \text{trace}(D^{-2} \chi^T D^2 \chi)/n_C \in (0, 1]\) over \( n_c \) to find the optimal number of clusters. In practice one may fall back to the heuristic initial guess for finding the optimal cluster number first before running the whole optimization for that number. Note that the optimization method for the final cluster number can be chosen differently from the method used to find the optimal cluster number. Typically, the Nelder-Mead method, which is fast but less accurate, is used for finding the optimal cluster number, whereas the slower but more accurate Gauss-Newton method is used for the final optimization.

In [10, 13], it is shown that the invariant sub-space condition

\[
 PX = \Lambda X, \quad \text{with} \quad \Lambda \in \mathbb{C}^{n \times n_C} \tag{4}
\]

in combination with the linear transformation \( \chi = XA \) and the orthonormality relation ensures the existence of a coarse-grained propagator matrix that preserves the dominant eigenvalues and hence the time scale of the slow processes. This coarse-grained matrix is defined as

\[
 P_c := (\chi^T D^2 \chi)^{-1} \chi^T D^2 P \chi. \tag{5}
\]

In fact, it holds

\[
P_c = A^{-1} \Lambda A,
\]

and hence

\[
(P^T_c)^k \chi^T \eta = \chi^T (P^T)^k \eta,
\]

which means that the propagation of the projected density vector (left) commutes with the projection of the propagated density \( \eta \) (right).

Interestingly, any linear transformation of the invariant subspace \( X \) preserves both the invariant subspace condition as well as the property of the column vectors to be almost constant on the blocks of a nearly decomposable Markov chain (because identical rows in \( X \) will also be identical after a linear transformation). Hence, the invariant subspace \( X \) does not necessarily need to be spanned by the eigenvectors. Any other invariant subspace, e.g. the Schur vectors, could be used as well in PCCA+.

### 2.1. G-PCCA+ algorithm

In order to avoid complex-valued eigenvectors and eigenvalues, which might occur for non-reversible Markov chains, the authors in [13] suggest to work with the real Schur-decomposition of \( P \). The resulting algorithm is named generalized PCCA+ (G-PCCA+).

After rearranging the real Schur-matrix \( \Lambda \) with the method described in [14], such that the \( n_c \) Schur-values are close to the Perron root 1, the authors scale the corresponding Schur-vectors by \( D_{\mu}^{-1/2} \), where \( D_{\mu} \) is a diagonal matrix with an arbitrary density \( \mu \) on its diagonal. They show that the scaled Schur vectors satisfy both the invariant sub-space...
condition and orthogonality relation. G-PCCA+ then applies the original PCCA+ algorithm to the scaled Schur-vectors to solve the optimization problem (3) in order to find the transformation matrix \( A \). Finally, G-PCCA+ calculates the coarse-grained propagation matrix \( P_c \) identical to (5) as

\[
P_c = (X^T D_\mu^2 X)^{-1} X^T D_\mu^2 P X.
\]

(6)

If the dominant eigenvalues are real numbers and if PCCA+ uses the same density \( \mu \) for the orthonormalization of eigenvector as G-PCCA+, then the two methods should deliver the same result.

2.2. Complex-valued PCCA+ (cPCCA+)

Instead of using the Schur-decomposition to deal with complex eigenvectors, we suggest to simply separate the real and imaginary parts of the eigenvectors. To show that this complex-valued version of PCCA+ (cPCCA+) actually results in an appropriate clustering of the system states, we are left to proof that the invariant sub-space condition holds. The following theorem shows that the matrix containing the real and complex parts of the eigenvectors is in fact a linear transformation of the original eigenvector matrix \( X \), making it again an invariant subspace of \( P \).

**Theorem 1.** (Invariant sub-space condition for complex eigenvectors) Replacing a complex conjugate pair of eigenvectors in the eigenvector matrix by the real and imaginary parts of these eigenvectors leads to a matrix that spans the same invariant subspace as the original eigenvector matrix. 

**Proof of Theorem 1.** Let \( X \in \mathbb{C}^{N \times n} \) be the matrix of eigenvectors, and assume that columns \( k \) and \( k+1 \) contain a complex conjugate pair of eigenvectors \( x_k = a_k + b_k i \) and \( \bar{x}_k = a_k - b_k i \) with \( a_k, b_k \in \mathbb{R}^N \) and imaginary unit \( i \). By replacing the columns \( x_k \) and \( \bar{x}_k \) by \( a_k \) and \( b_k \), we obtain a new matrix \( \tilde{X} \).

We have to show that the matrix \( \tilde{X} \) can be obtained from \( X \) via a linear transformation, i.e., via multiplication with a regular matrix \( C \),

\[
\tilde{X} = X \cdot C.
\]

A transformation matrix \( C \in \mathbb{R}^{N \times N} \) that only acts on columns \( k \) and \( k+1 \) is the identity matrix with some additional \( 2 \times 2 \) block on the diagonal corresponding to rows and columns \( k \) and \( k+1 \):

\[
C = \begin{pmatrix}
1 & 0 & \ldots & 0 \\
0 & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
c_1 & c_2 & \ddots & 1 \\
0 & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
c_3 & c_4 & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots \\
\end{pmatrix}
\]

In terms of columns \( k \) and \( k+1 \), the system of equations \( \tilde{X} = X \cdot C \) reads

\[
\begin{bmatrix}
a_k & b_k \\
a_k + b_k i & a_k - b_k i
\end{bmatrix} = \begin{bmatrix}
a_k + b_k i & a_k - b_k i
\end{bmatrix} \times \begin{bmatrix}
c_1 & c_2 \\
c_3 & c_4
\end{bmatrix}.
\]

(7)

Hence, we have to find values for \( c_1, \ldots, c_4 \), such that

\[
c_1(a_k + b_k i) + c_2(a_k - b_k i) = a_k,
\]

\[
c_3(a_k + b_k i) + c_4(a_k - b_k i) = b_k.
\]

This system of equations has the unique solution \( c_1 = c_2 = \frac{1}{2} \) and \( c_3 = -\frac{1}{2} i = -c_4 \). Hence, the \( 2 \times 2 \) matrix formed by \( c_1 \) to \( c_4 \) and consequently \( C \) is regular. The linear combination \( X \cdot C \) is thus an invariant sub-space of \( P \) again. \( \square \)

Theorem 1 implies that the cPCCA+ algorithm is able to cluster states in a system with complex dominant eigenvalues. This allows us to conclude that G-PCCA+ and cPCCA+ are equivalent in the following sense:

**Corollary 1.** If \( P \) admits an eigen-decomposition, G-PCCA+ and cPCCA+ obtain the same solutions to the optimization problem.
Spectral clustering of Markov chains with complex eigenvalues

Figure 1: Weighted directed graph for the Markov chain in Example 1.

PROOF. Both algorithms solve the original PCCA+ problem (1-3) with different bases $X_1, X_2$ of the same subspace span$X_1 = $ span$X_2$. Thus there exists a regular matrix $B$ s.t. $X_1 = X_2B$. Then any solution to the objective function of the form $\chi = X_1A_1$ is also a solution of the form $\chi = X_2BA_1 = X_2A_2$ with $A_2 = BA_1$ and vice versa. □

Note that in practice we use only local optimization and hence the different representations and resulting initial guesses may lead to different local minima which, however, are also local minima to the respective other problem. We furthermore point out that there exist non-generic transition matrices $P$ which admit no complex eigen-decompositions but can still be handled by the Schur method [17].

In the following, we will consider examples for the case of (i) a nearly uncoupled Markov chain, and (ii) a Markov chain with circular transition pattern, and argue in both cases for the fulfillment of the invariant subspace condition.

2.2.1. Non-reversible, nearly uncoupled Markov chains

In Example 1, we consider the case of a small, non-reversible system with three metastable states and show that cPCCA+ clusters the states accordingly.

Example 1. Let $P \in \mathbb{R}^{6 \times 6}$ be the transition matrix for the Markov chain in Fig. 1:

$$P = \begin{bmatrix}
0.25 & 0.7 & 0 & 0 & 0 & 0.05 \\
0.70 & 0.29 & 0.01 & 0 & 0 & 0 \\
0 & 0.05 & 0.25 & 0.70 & 0 & 0 \\
0 & 0 & 0.70 & 0.29 & 0.01 & 0 \\
0 & 0 & 0 & 0.05 & 0.25 & 0.70 \\
0.01 & 0 & 0 & 0 & 0.70 & 0.29
\end{bmatrix}$$

This matrix has three eigenvalues close to one: $\lambda_1 = 1.0000, \lambda_2 = 0.9557 + 0.0177i, \lambda_3 = 0.9557 - 0.0177i$. Figure 2 shows the fuzzy clustering of the six states into three clusters according to the membership functions $\chi$ computed by cPCCA+. Each of the three colors (blue, red and yellow) represents a different cluster. The system states with values of $\chi$ near 1, are those that strongly belong to a specific cluster. The resulting coarse-grained transition matrix $P_c \in \mathbb{R}^{3 \times 3}$ in (10) has diagonal-entries close to unity, which indicates three strong metastable clusters.

$$P_c = \begin{bmatrix}
0.9705 & 0.0046 & 0.0250 \\
0.0250 & 0.9705 & 0.0046 \\
0.0046 & 0.0250 & 0.9705
\end{bmatrix}$$

This example illustrates that cPCCA+ can deal with non-reversible, stochastic matrices with dominant complex eigenvalues. In the next section, we consider circular transition patterns.
Spectral clustering of Markov chains with complex eigenvalues

Figure 2: Membership functions $\chi$ for Example 1: The functions show that the six system states are clustered into three distinct groups, each containing two states.

### 2.2.2. Circular Markov chains

In this sub-section, we make use of the unique structure of circular matrices. In Theorem 2, we show that the invariant sub-space condition holds for circular stochastic matrices, which implies that cPCCA+ can be applied to such matrix structures.

**Theorem 2.** (Invariant sub-space condition for circular transition matrices) Let $P \in \mathbb{R}^{(N \times n) \times (N \times n)}$ be a row-stochastic, circular matrix with the following structure,

$$P = \begin{bmatrix} 0 & X_1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & X_{N-1} \\ X_N & 0 & \cdots & 0 \end{bmatrix},$$

where $X_k \in \mathbb{R}^{n \times n}$, $k = 1, \ldots, N$ are row stochastic sub-matrices. Then the matrix $P$ has $N$ eigenvalues $\lambda_1, \ldots, \lambda_N \in \mathbb{C}$ uniformly distributed on the unit circle,

$$\lambda_k = \exp((k - 1) \theta i) \text{ with } \theta = \frac{2\pi}{N}, \quad k = 1, \ldots, N.$$

The corresponding eigenvectors $V_k$, i.e., those satisfying $PV_k = \lambda_k V_k$ for $k = 1, \ldots, N$, are piece-wise constant on each block $X_k$.

**Proof of Theorem 2.** The eigenvectors $V_k$ corresponding to eigenvalues $\lambda_k$ can be written in a general form as

$$V_k^T = \begin{bmatrix} \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \\ \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \\ \vdots & \ddots & \ddots & \vdots \\ \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \\ \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \\ \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \\ \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \\ \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \\ \lambda_k^1, \lambda_k^2, \ldots, \lambda_k^N \end{bmatrix}^T.$$

Noting that $\lambda_k \cdot \lambda_k^N = \lambda_k^{N+1} = \lambda_k$ and that the matrix $P$ shifts the process in clockwise direction to the next block, it holds, that

$$P \cdot V_k = \begin{bmatrix} \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \\ \lambda_k^2, \lambda_k^3, \lambda_k^4, \ldots, \lambda_k^N \end{bmatrix}^T = \lambda_k \cdot V_k,$$

which proves the eigenvectors corresponding to eigenvalues on the unit circle are piecewise constant on the blocks. $\square$

Next, we demonstrate that cPCCA+ can still be applied to a non-reversible transition matrix $P \in \mathbb{R}^{9 \times 9}$ where the circular transition pattern is slightly perturbed. The matrix structure in Example 2 is taken from [13] (see Example 3.1 therein).

**Example 2.** We consider a stochastic matrix $P \in \mathbb{R}^{9 \times 9}$,

$$P = \begin{bmatrix} X & Y & 0 \\ 0 & X & Y \\ Y & 0 & X \end{bmatrix}.$$
Spectral clustering of Markov chains with complex eigenvalues

| Matrix entries for x and y: | Dominant eigenvalue type: |
|---------------------------|--------------------------|
| x = 0.9, y = 0.1          | Case (i)                 |
| x = 0.1, y = 0.9          | Case (ii)                |
|                           | Case (iii)               |
|                           | Case (iv)                |

Table 1
The four cases of matrix patterns in Example 2

Consisting of identical diagonal blocks \( X \in \mathbb{R}^{3 \times 3} \) with the structure

\[
X = \begin{bmatrix}
0 & x & 0 \\
0 & 0 & 1.0 \\
1.0 & 0 & 0
\end{bmatrix},
\]

(14)

and identical off-diagonal blocks \( Y \in \mathbb{R}^{3 \times 3} \) with the following pattern,

\[
Y = \begin{bmatrix}
y & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}.
\]

The \( 0 \in \mathbb{R}^{3 \times 3} \) represent zero block matrices.

The matrix \( P \) contains one block-diagonal pattern, formed by the \( X \) blocks in \( P \), and additionally two types of circular patterns, one within each \( X \) block and one formed by the \( Y \) blocks in \( P \). Whether one identifies the block-diagonal or the circular pattern depends on the values of \( x \in \mathbb{R} \) and \( y \in \mathbb{R} \), as well as on the eigenvalues that are considered as dominant (either the ones with largest magnitude or those with largest real part). In the following, we analyze the four different cases (i)-(iv), as presented in Table 1: We examine the eigenvalues, the membership functions \( \chi \), and the coarse-grained matrices \( P_c \) computed by cPCCA+ in each of the four cases.

**Case (i).** The three eigenvalues with largest magnitude are \( \lambda_1 = 1.0000 + 0i, \lambda_2 = -0.5000 + 0.8660i, \lambda_3 = -0.5000 - 0.8660i \). Figure 3 shows the clustering of the nine states into three clusters based on the \( \chi \) membership functions. The three clusters are characterized by sharing the same transition pattern: cluster 1 (blue - states 1, 6, 8) transitions to cluster 2 (red - states 2, 4, 9), cluster 2 transitions to cluster 3 (yellow - states 3, 5, 7), and cluster 3 transitions back to cluster 1. That means the symmetry in the transition graph is perfectly recovered. The resulting coarse-grained transition matrix reads

\[
P_c = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}.
\]

(15)

**Case (ii).** If we instead solve for the eigenvalues with largest (real) part, we obtain \( \lambda_1 = 1.0000 + 0i, \lambda_2 = 0.9483 + 0.0279i, \lambda_3 = 0.9483 - 0.0279i \), indicating that the Markov chain represents a nearly-uncoupled Markov chain. Figure 3 shows the clustering of the nine states into three clusters based on the \( \chi \) membership functions. The three clusters reflect the block diagonal structure that is visible in \( P \): cluster 1 includes states 1, 2, 3 (blue), cluster 2 includes states 4, 5, 6 (red), and cluster 3 includes states 7, 8, 9 (yellow). The resulting coarse-grained transition matrix has the form

\[
P_c = \begin{bmatrix}
0.9655 & 0.0333 & 0.0012 \\
0.0012 & 0.9655 & 0.0333 \\
0.0333 & 0.0012 & 0.9655
\end{bmatrix}.
\]

(16)

which reflects the decoupling of the chain into three nearly decoupled sub-chains and the perfect symmetry between the clusters. For \( x = 1 \) and \( y = 0 \), the decoupling would be complete and \( P_c \) would be the identity matrix with zero transition probabilities between the clusters.

**Case (iii).** For \( x = 0.1 \) and \( y = 0.9 \), i.e., an increased coupling between the sub-chains represented by \( X \), the three eigenvalues with largest magnitude are (upon permutation) \( \lambda_1 = 1.0000 + 0i, \lambda_2 = -0.5000 + 0.8660i, \lambda_3 = -0.5000 - 0.8660i \), indicating that there is a circular transition pattern. Figure 3 shows that cPCCA+ identifies the
Spectral clustering of Markov chains with complex eigenvalues

Case (i). $x = 0.9$, $y = 0.1$, largest magnitude eigenvalue
Three circulating clusters

Case (ii). $x = 0.9$, $y = 0.1$, largest (real) eigenvalue
Three block-group clusters

Case (iii). $x = 0.1$, $y = 0.9$, largest magnitude eigenvalue
Three circulating clusters

Case (iv). $x = 0.1$, $y = 0.9$, largest (real) eigenvalue
No multistability, hence no clear clustering

Figure 3: Example 2: Membership functions $\chi$ (left column) and transition networks (right column)

same circular transition pattern as in case (i), except that clusters 2 and 3 are permuted\(^1\). That means, cluster 1 (blue - states 1,6,8) transitions into cluster 3 (yellow - states 2,4,9), cluster 3 transitions into cluster 2 (red - states 3,5,7),

\(^1\)The clustering delivered by PCCA+ is unique up to a permutation of membership vectors.
and cluster 2 transitions back into cluster 1:

\[
P_c = \begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}.
\] (17)

Case (iv). Applying cPCCA+ to Case (ii) specifying largest (real) eigenvalue, results in only one eigenvalue close to 1 (\(\lambda_{2,3} = 0.2954 \pm 0.1128i\)), i.e., there is no multi-stability. If we nevertheless enforce a clustering into three groups, then states 2, 5, and 8 form the centers of these three clusters, because the transition probabilities between these three states are very small. States 1, 4, and 7 are assigned to the three clusters with almost equal membership, indicating that they are transition states between the three clusters. Each of the states 3, 6, and 9 is assigned to one of the three clusters with a membership value > 0.5, which shows that they are stronger associated with the clusters than the states 1, 4, and 7. The coarse-grained transition matrix (for cluster order blue - red - yellow) has the form

\[
P_c = \begin{bmatrix}
0.5303 & 0.3000 & 0.1697 \\
0.1697 & 0.5303 & 0.3000 \\
0.3000 & 0.1697 & 0.5303 \\
\end{bmatrix}.
\] (18)

which underlines that the three clusters are only weakly decoupled.

Above, we have shown theoretically and numerically that the invariant sub-space condition holds for the cPCCA+ algorithm, when applied to non-reversible transition matrices with complex eigenvalues, including circuit stochastic matrices. These findings were supported by specific examples. In the following, we want to compare the numerical runtimes and clustering results of cPCCA+ and G-PCCA+ for two larger matrix examples.

3. Comparison of numerical runtimes

To compare the numerical runtimes and clustering results of cPCCA+ and G-PCCA+, we consider the cases of (1) a circular transition matrix (taken and adapted from [13]; see Example 3.2 therein), and (2) the infinitesimal generator of a stochastic, non-reversible gene-regulatory network, generated from the model in [18].

For both examples, we compare the computational elapse time (ET) for varying state-space sizes as well as the normed differences between the coarse-grained transition matrices \(P_c\) obtained from cPCCA+ and G-PCCA+, respectively. The ET was calculated with the \(\text{tic-toc}\) command in Matlab (version '9.7.0.1261785 (R2019b) Update 3'), and the normed differences were determined with the \(\text{norm(difference, } p)\) function in Matlab (version ‘9.7.0.1261785 (R2019b) Update 3’) with \(p \in \{1, 2, \infty\}\).

In the following, we present the generation process of the example matrices and the results of the comparison for the two cases.

1. Circular transition patterns with and without perturbations. In this case, we analyze circular matrices \(P\) with varying state-space size \(N\) for \(N = 30, 60, 90, 120\). To construct the matrices \(P\), we proceed similar to Example 3.2 in [13]: We start by creating three n-by-n random matrices A1, A2, and A3 using the Matlab-routine \(\text{rand(n,n)}\) (MATLAB version ‘9.7.0.1261785 (R2019b) Update 3’), and one n-by-n zero matrix Z, where \(n\) takes the values 10, 20, 30 and 40. A circular matrix structure is attained by defining another matrix C, via C:=\([Z,A1,Z];[Z,Z,A2];[A3,Z,Z]\], which varies in state-space size as specified above. Upon re-scaling, matrix C becomes stochastic and represents the non-perturbed circular matrices \(P\) to be analyzed. In order to create perturbed stochastic circular matrices \(P\), a random matrix containing entries between 0 and 0.1 is added to matrix C before it is re-scaled.

To account for variability in the random generation of the circular matrices and to estimate the mean and standard deviation of ETs and clustering results, we generate, for each matrix size \(N\) and each case (perturbed and non-perturbed), five different matrices \(P\) (according to the steps described above).

The results show that both in the non-perturbed case (Fig. 4) and in the perturbed case (Fig. A.1), cPCCA+ and G-PCCA+ are fast, but that cPCCA+ is consistently quicker.

We also see in Figures 5 (non-perturbed case) and A.2 (perturbed case) that the differences in the coarse-grained transition matrices are small, although it seems that the difference in the perturbed case increases with matrix size (Figure A.2).
2. **Stochastic gene-regulatory network.** We transformed the 2-dimensional deterministic macrophage polarization model described in [18] into a stochastic gene-regulatory network whose dynamic is described by the Chemical Master Equation (CME). We analyze the system on a finite state space of size $N^2$ for $N \in \{20, 40, 60, 80, 100\}$. For the considered parameter set, the system shows tri-stability.

Making use of the calculated ET for the cPCCA+ and G-PCCA+, we determine the procentual difference in ETs between these two methods, to which we fit a quadratic polynomial curve ($f(x) = ax^2 + bx + c$). This approach allows us to quantify a functional relationship between the runtimes of both methods. In addition, we aim to identify the G-PCCA+ sub-routine in which the algorithm spends most time. To this end, we determine ETs for different subroutines in the G-PCCA+ algorithm. These sub-routines are the minChi optimization routine, optimization procedures with methods 'nelder-mead' (optimization loop 1) and 'gauss-newton' (optimization loop 2), the Schur-decomposition routine, and the sorting of the real Schur-vectors.

As the state space increases, we observe a quadratic growth in the difference of ETs between cPCCA+ and G-PCCA+ (see Figures 6 and B.1), while the difference between the coarse-grained transition matrices remains small (see Figure 7).

In addition, we could allocate that the G-PCCA+ approach spends at least half of the total ET performing the Schur-decomposition and sorting the Schur-vectors (see Figure B.2), especially for large $N$.

4. Discussion

In this article, we have introduced a computationally efficient (fast and accurate) algorithm, named cPCCA+, for coarse-graining of large, non-reversible stochastic matrices in Matlab. cPCCA+ is equal to the original PCCA+ method by [10] but applies a separation of real and imaginary eigenvectors. We have justified the approach based on theoretical derivations and tested it on particular examples. We compared the results from cPCCA+ with the G-PCCA+ method by [13] on two large matrix examples. Based on the presented results, we could show that the cPCCA+ algorithm
Spectral clustering of Markov chains with complex eigenvalues

Figure 6: Elapse times (ET) of c-PCCA+ and G-PCCA+ for the stochastic gene-regulatory network matrix with increasing state space $N^2$.

Figure 7: Normed error differences in coarse-grained transition matrices $P_c$ calculated with cPCCA+ and G-PCCA+ for the stochastic gene-regulatory network with increasing state space $N^2$.

is faster, but that the results are equally accurate as the results of the G-PCCA+ algorithm, which was confirmed by Corollary 1.

The importance of the presented algorithm can be acknowledged by considering the stochastic nature of cellular phenotype switches. One prominent example are cancer cells in tumors, which exhibit a diverse phenotype spectrum, and whose transition patterns are assumed to follow a Markov model, see e.g., [19]. The original PCCA+ algorithm [10] has previously been proven valuable for the clustering of Markov chains that describe gene regulatory networks [1, 2], with its limitation to be only applicable to reversible systems. Our adaptations in cPCCA+ make it now possible to analyze more general systems of, e.g., gene regulatory networks that exhibit non-reversible transition patterns.

As demonstrated in this article, both the cPCCA+ and G-PCCA+ algorithm are suitable for dealing with cases of large non-reversible matrices in Matlab. However, cPCCA+ has been proven to overcome the computational burden that is attributable to the Schur-decomposition and the sorting of Schur-vectors in G-PCCA+.

While we have focused in this article on the PCCA+ algorithms for the Matlab programming language, we would like to point out that the G-PCCA+ algorithm is also implemented in the Python packages pyGPCCA [20] and cmdtools [21]. The latter introduces the Krylov-Schur-decomposition [22] from the PETSc / SLEPc library [23], which iteratively computes the leading Schur-vectors and therefore overcomes the computational costs of the direct solve and sort routine of the Matlab code.

5. Conclusion

Upon a separation of the real and imaginary parts of complex eigenvectors, cPCCA+ expands the original PCCA+ algorithm, and makes it applicable to large, non-reversible stochastic matrices, including circular transition matrices. Such matrices often occur in real-world applications, such as gene-regulatory networks. The algorithm is faster, but as accurate as the current state-of-the-art approach (e.g., the G-PCCA+ method). This makes cPCCA+ an efficient alternative to the G-PCCA+ algorithm in the Matlab programming language.
A. Additional figures for the perturbed circular matrix example (Section 3.1)

Figure A.1: Elapse Time (ET) of c-PCCA+ and G-PCCA+ for perturbed circular matrix. The figure shows an error bar plot with the mean and standard deviation of five ETs for each state space size $N$.

Figure A.2: Normed difference in coarse-grained transition matrices $P_c$ for five perturbed circular matrices per state space size $N$. 

| State space size $N$ | Elapse time (ET) in min (mean ± std) |
|----------------------|--------------------------------------|
| 30                   | 0.08 ± 0.00                          |
| 60                   | 0.06 ± 0.00                          |
| 90                   | 0.05 ± 0.00                          |
| 120                  | 0.04 ± 0.00                          |

Frank et al.: Preprint submitted to Elsevier
B. Additional figures for the gene-regulatory network example (Section 3.2)

Figure B.1: Quadratic polynomial curve fit to percental differences in ET between cPCCA+ and G-PCCA+ for the gene-regulatory network example for increasing state space $N^2$.

Figure B.2: Cumulative plot of elapse times (ET) for different sub-routines in G-PCCA+ for increasing state-space $N^2$. Sub-routines are: **minChi criterion**: Determines potential range of numbers of meta-stable states $n_c$; **SRSchur**: Schur-vectors are sorted/re-arranged with respect to their distance form the unit circle; **Optimization loop**: The optimization loop for finding the transformation matrix $A$ is a two-step process. In the first step, the algorithm uses the ‘nelder-mead’ optimization routine in order to find the optimal cluster number $n_c$ (Iteration 1). In the second step, the ‘gauss-newton’ routine is used to iteratively optimize $A$ for the optimal cluster number $n_c$ (Iteration 2). **Schur decomposition**: Schur decomposition calculates all Schur vectors of matrix $P$. For more information on the different sub-routines, see [8].
Data statement

The cPCCA+ Matlab codes for the examples 1 and 2 can be found on GitHub (https://github.com/sroeblitz/cPCCA). The G-PCCA+ Matlab code, as described in [8], was downloaded from GitHub (https://github.com/msmdev/gpcca) and adapted to our examples.

Funding sources

The work of AF and SR was supported by Trond Mohn Foundation (BSF, https://www.mohnfoundation.no/), grant no. BFS2017TMT01. The work of AS has been funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 “Scaling Cascades in Complex Systems”, Project Number 235221301, Project A05 “Probing scales in equilibrated systems by optimal nonequilibrium forcing”.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Anna-Simone Frank: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. Alexander Sikorski: Methodology, Validation, Investigation, Writing - review & editing. Susanna Röblitz: Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization, Supervision.

References

[1] B.K. Chu, M.J. Tse, R.R. Sato, and E.L. Read. Markov State Models of gene regulatory networks. BMC Syst Biol, 11(14):1–17, 2017. https://doi.org/10.1186/s12918-017-0394-4.
[2] M.J. Tse, B. K. Chu, C.P. Gallivan, and E.L. Read. Rare-event sampling of epigenetic landscapes and phenotype transitions. PLoS Comput Biol, 14(8):e1006336, 2018. https://doi.org/10.1371/journal.pcbi.1006336.
[3] W.C. Swope, J.W. Pitera, F. Suits, M. Pitman, M. Eleftheriou, B.G. Fitch, R.S. Germain, A. Rayshubski, T.J.C. Ward, Y. Zhestkov, and R. Zhou. Describing protein folding kinetics by molecular dynamics simulations. 2. example applications to alanine dipeptide and a β-hairpin peptide. J Phys Chem B, 108(21):6582–6594, 2004. https://doi.org/10.1021/jp037422q.
[4] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T.R. Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. PNAS, 106(45):19011–19016, 2009. https://doi.org/10.1073/pnas.0905466106.
[5] A.R. Fersht. Characterizing transition states in protein folding: an essential step in the puzzle. Curr Opin Struct Biol, 5(1):79–84, 1995. https://doi.org/10.1016/0959-440X(95)80012-P.
[6] J.D. Chodera and F. Noé. Markov state models of biomolecular conformational dynamics. Curr Opin Struct Biol, 25:135–144, 2014. https://doi.org/10.1016/j.sbi.2014.04.002.
[7] P.E.P. Burke, C.B.d.L. Campos, L.d.F. Costa, and M.G. Quiles. A biochemical network modeling of a whole-cell. Sci Rep, 10(13303):1–14, 2020. https://doi.org/10.1038/s41598-020-70145-4.
[8] B. Reuter, M. Weber, K. Fackeldey, S. Röblitz, and M.E. Garcia. Generalized Markov State Modeling Method for Nonequilibrium Biomolecular Dynamics: Exemplified on Amyloid β Conformational Dynamics Driven by an Oscillating Electric Field. J Chem Theory Comput, 14(7):3579–3594, 2018. https://doi.org/10.1021/acs.jctc.8b00079.
[9] V.S. Pande, K. Beauchamp, and G.R. Bowman. Everything you wanted to know about Markov State Models but were afraid to ask. Methods, 52(1):99–105, 2010. https://doi.org/10.1016/j.meth.2010.06.002.
[10] S. Andrilli and D. Hecker. Chapter 8 - Additional Applications. In Elementary Linear Algebra, pages 491–585. Academic Press, Boston, Fourth edition, 2010. https://doi.org/10.1080/00029890.2018.1468609.
[11] J.H. Brandts. Matlab code for sorting real Schur forms. Numer Linear Algebra Appl, 9(3):249–261, 2002. https://doi.org/10.1002/nla.274.
[12] M. Weber. Implications of PCCA+ in Molecular Simulation. Comput, 6(1):20, 2018. https://doi.org/10.3390/computation6010020.
[13] P. Deuflhard and M. Weber. Robust Perron cluster analysis in conformation dynamics. Linear Algebra Appl, 398:161–184, 2005. https://doi.org/10.1016/j.laa.2004.10.026.
[17] Marcus Weber. Eigenvalues of non-reversible Markov chains - A case study. Technical Report 17-13, ZIB, 2017. Available online at http://nbn-resolving.de/urn:nbn:de:0297-zib-62191; Accessed on June 14, 2022.

[18] A.S. Frank, K. Larripa, H. Ryu, R.G. Snodgrass, and S. Röblitz. Bifurcation and sensitivity analysis reveal key drivers of multistability in a model of macrophage polarization. *J Theor Biol*, 509(110511), 2021. https://doi.org/10.1016/j.jtbi.2020.110511.

[19] P.B. Gupta, C.M. Fillmore, G. Jiang, S.D. Shapira, K. Tao, C. Kuperwasser, and E.S. Lander. Stochastic State Transitions Give Rise to Phenotypic Equilibrium in Populations of Cancer Cells. *Cell*, 146(4):633–644, 2011. https://doi.org/10.1016/j.cell.2011.07.026.

[20] B. Reuter, M. Klein, and M. Lange. pyGPCCA – python GPCCA: Generalized Perron Cluster Cluster Analysis package to coarse-grain reversible and non-reversible Markov State Models. GitHub, 2021. Available online at https://github.com/msmdev/pyGPCCA.

[21] A. Sikorski, R. Sechi, and L. Helfmann. cmdtools: Python implementation of several tools for the analysis of dynamical systems from the transfer operator perspective. GitHub, 2021. https://doi.org/10.5281/zenodo.4749330; Available online at https://github.com/zib-cmd/cmdtools/tree/v1.0.1.

[22] G.W. Stewart. A Krylov-Schur Algorithm for Large Eigenproblems. *SIAM J Matrix Anal Appl*, 23(3):601–614, 2001. https://doi.org/10.1137/S0895479800371529.

[23] V. Hernández, J.E. Román, A. Tomás, and V. Vidal. Krylov-Schur methods in SLEPc. Slepc technical report str-7, Scalable Library for Eigenvalue Problem Computations, Universitat Politécnica de València, June 2007. https://slepc.upv.es/documentation/reports/str7.pdf.