

# Language in Time - Introduction

This short manual is designed primarily for researchers in the social sciences who work with text, transcripts, or related data, and wish to think about text in terms of dynamical systems. The manual is focused on the quantification and visualization of the temporal organization of these data sources. I will also describe the relationships between the approaches of this manual to techniques in computational linguistics, natural language processing, and other statistical domains. Recurrence quantification has conceptual and computational overlap with many techniques in those domains.

In this first section, we introduce the rudiments. Starting with some very simple text data, we will showcase how to import, convert, and plot some simple aspects of raw text. We then introduce the recurrent plot (RP).

In the second section, we introduce quantification of recurrence plots, called recurrence quantification analysis (CRQA). We will describe various measures to you, and identify how the measures are related to each other.

In the third section, we cover cross-recurrence quantification analysis (CRQA) and describe the corresponding quantities that accompany it.

In the fourth section, we describe windowed (C)RQA, which allows us to explore how these measures change over time in a text or related data set.

In the last section, we describe how (i) many of the measures of recurrence relate to quantities in NLP and related domains (such as n-gram models) and (ii) we show how (C)RQA can be adapted for the purpose of getting other NLP measures to describe a text in time. The overall goal is not to argue that recurrence is better or worse – but rather it is a descriptive-statistic framework that allows researchers to think in terms of time.

This manual uses the `crqa` library written in R primarily by Dr. Moreno Coco with a modicum of help from me, and with considerable inspiration from the impressive dynamics group at the University of Cincinnati (Jay Holden, Mike Richardson, Kevin Shockley, Mike Riley, and Guy Van Orden).