



**HAL**  
open science

# Humanités numériques et computationnelles appliquées à l'étude de l'écrit ancien

Peter Anthony Stokes

► **To cite this version:**

Peter Anthony Stokes. Humanités numériques et computationnelles appliquées à l'étude de l'écrit ancien. Annuaire de l'École pratique des hautes études. Section des sciences historiques et philologiques, 2021, 152, pp.506-508. 10.4000/ashp.4918 . hal-03991749

**HAL Id: hal-03991749**

**<https://hal.science/hal-03991749v1>**

Submitted on 16 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

## Humanités numériques et computationnelles appliquées à l'étude de l'écrit ancien

Peter A. Stokes

---



### Édition électronique

URL : <https://journals.openedition.org/ashp/4918>

DOI : [10.4000/ashp.4918](https://doi.org/10.4000/ashp.4918)

ISSN : 1969-6310

### Éditeur

Publications de l'École Pratique des Hautes Études

### Édition imprimée

Date de publication : 1 septembre 2021

Pagination : 506-508

ISSN : 0766-0677

### Référence électronique

Peter A. Stokes, « Humanités numériques et computationnelles appliquées à l'étude de l'écrit ancien », *Annuaire de l'École pratique des hautes études (EPHE), Section des sciences historiques et philologiques* [En ligne], 152 | 2021, mis en ligne le 14 juin 2021, consulté le 18 juin 2022. URL : <http://journals.openedition.org/ashp/4918> ; DOI : <https://doi.org/10.4000/ashp.4918>

---

## HUMANITÉS NUMÉRIQUES ET COMPUTATIONNELLES APPLIQUÉES À L'ÉTUDE DE L'ÉCRIT ANCIEN

Directeur d'études : M. Peter A. STOKES

Programme de l'année 2019-2020 : *Approches numériques et computationnelles du polygraphisme.*

Malgré une année fortement perturbée, nous avons continué nos travaux sur la modélisation des écritures. Notre problématique était à nouveau de savoir comment, et dans quelle mesure, les méthodes existantes développées pour la paléographie latine peuvent être étendues à d'autres systèmes d'écriture. Cette année nous nous sommes concentrés sur l'écriture des inscriptions en vieux khmer du IX<sup>e</sup> et X<sup>e</sup> siècles de notre ère.

En commençant avec une reprise des problématiques, nous avons fait des expériences avec le logiciel Archetype<sup>1</sup>. La première étape a consisté à examiner le modèle existant et à voir dans quelle mesure celui-ci peut s'appliquer aux inscriptions khmères<sup>2</sup>. Le modèle d'Archetype possède suffisamment de généralité pour être appliqué à presque tous les systèmes d'écriture. La pratique est, bien entendu, toujours plus complexe et il est inévitable que certaines caractéristiques propres à l'écriture des inscriptions en vieux khmer ne soient pas prises en compte. De même, certains éléments du modèle considérés comme essentiels pour la paléographie latine sont sans intérêt ou sans grande signification dans d'autres contextes. C'est le cas par exemple, de la distinction entre lettres majuscules et minuscules. Malgré ces quelques différences, le modèle initial paraît toujours pertinent. Comme c'est toujours le cas lors de la modélisation de matériaux de recherche dans des contextes numériques, nous avons fait face à des questions d'ordre épistémologiques. En raison des traces plus longues et souvent plus complexes dans ce système d'écriture par rapport au latin, les composantes significatives de l'écriture furent souvent difficile à définir. En effet, l'écriture khmer de cette période est plus proche de ce que Collette Sirat qualifiait d'« écritures de ligne » (*line script*) alors qu'Archetype a été conçu pour étudier les « écritures de traits » (*stroke script*)<sup>3</sup>. Une autre difficulté particulière est la « marque de tête » que l'on trouve dans l'écriture khmère et qui n'a pas d'équivalent direct dans les systèmes d'écriture d'Occident. Sa fonction n'est d'ailleurs pas tout

1. S. Brookes, P. A. Stokes, M. Watson, Débora Marques de Matos, « The DigiPal Project for European Scripts and Decorations », dans A. Conti, O. da Rold et P. Shaw (éd.), *Writing Europe 500–1450: Texts and Contexts*, Cambridge, D. S. Brewer, 2015, p. 25-58.
2. Pour la modélisation dans les humanités numériques, voir P. A. Stokes, « Humanités numériques et computationnelles appliquées à l'étude de l'écrit ancien », *Annuaire. Résumés des conférences et travaux*, 151<sup>e</sup> année, 2018-2019, Paris, EPHE, PSL, SHP, 2020, p. 437-439.
3. C. Sirat, *Writing as Handwork: A History of Handwriting in Mediterranean and Western Culture*, Turnhout, Brepols, 2006, p. 314-315.

à fait claire, de sorte que sa catégorisation dans le modèle n'est nullement évidente. Toutefois, le modèle a pu être appliqué pour catégoriser un certain nombre de signes (*akṣara*), signes que nous avons ensuite utilisés pour annoter et analyser un ensemble d'inscriptions à l'aide d'Archetype.

L'étape suivante fut d'utiliser le logiciel et les données produites pour l'analyse. L'interface existante ayant été conçue pour les manuscrits occidentaux en latin, cette dernière n'était pas adaptée à notre corpus. Cette difficulté est liée à une autre tension inhérente aux outils numériques : plus une interface est facile à utiliser, moins elle est puissante et flexible. Les analyses entreprises par le chercheur sont toujours très variées et peuvent rarement être réduites à de simples opérations ; cela implique nécessairement un large éventail d'options et de détails différents que l'utilisateur doit maîtriser. Il est donc presque inévitable que tout logiciel développé pour la recherche en sciences humaines ne réponde pas entièrement aux besoins du chercheur, un certain degré de personnalisation est donc presque toujours nécessaire. Archétype (et de nombreuses plateformes similaires) offre deux moyens de remédier à ce problème : on peut soit en modifier directement le code source (l'un des nombreux arguments en faveur de logiciel en libre accès), soit utiliser une interface de programmation d'application (API), un procédé permettant d'interagir directement avec la base de données en utilisant son propre code informatique pour les cas où l'interface existante est insuffisante. Ces méthodes nécessitent une connaissance de base de la programmation, et c'est pourquoi nous avons passé un certain temps à voir les bases de Python, qui est le langage le plus utilisé pour les humanités numériques à l'heure actuelle. Python a l'avantage d'être relativement puissant et de permettre de réaliser des logiciels complexes de taille relativement importante ; il est aussi relativement facile à apprendre, et les débutants peuvent produire des logiciels simples mais utiles beaucoup plus rapidement qu'avec de nombreux autres langages de programmation. Par ailleurs, une grande quantité de ressources sont disponibles pour aider les étudiants à apprendre le langage, y compris des tutoriels ciblés spécifiquement aux chercheurs dans le domaine des sciences humaines<sup>4</sup>. De plus, Archétype est lui-même écrit presque entièrement en Python. Cela fait de Python le langage le plus naturel pour l'interaction avec Archétype. Après un certain temps d'apprentissage des bases, nous avons écrit des exemples simples en Python afin d'interagir directement avec la base de données et d'effectuer des requêtes personnalisées qui ne sont pas possibles avec l'interface existante. Ce langage de programmation est également utile pour automatiser des tâches répétitives. Nous l'avons notamment utilisé pour télécharger automatiquement un corpus d'images depuis l'Internet à l'aide du standard *International Image Interoperability Framework* (IIIF : <https://iiif.io>) ainsi que pour redimensionner les images et charger les données dans Archétype afin de créer un flux de travail plus efficace.

4. Voir, par exemple, *The Programming Historian*, <https://programminghistorian.org>.

