The *Journal of Multimodal Rhetorics*

Volume 5, Issue 2

JoMR

A Feeling for the Algorithm:
A Feminist Methodology for Algorithmic Writing and Research

Patricia Fancher, University of California, Santa Barbara

Abstract
I offer this narrative as a feminist methodology for algorithmic research methods, insisting that affect, collaboration, community and embodiment are rich resources for algorithmic writing and research. I highlight these feminist themes because much of the research on algorithmic writing focuses procedural rhetoric and the relationship between the programmer and the program. In my narrative, I highlight the social and embodied experiences that shape my process as a feminist working with algorithmic research methods. In doing, I resist the assumption that feminist scholars have a technical deficiency that must be overcome in order to engage in DH research. Instead, I position our feminist practices and values as vital assets for conducing compelling feminist research in digital humanities.

Keywords: Feminist; Digital Humanities; Methodology; Affect; Embodiment.

Introduction
Barbara McClintock is now famous for her scientific breakthroughs regarding genetic code. However, for 30 years, no one took her seriously. Genome sequences were, and largely still are, understood as mechanical, code, immutable, and 'objective' in so far as DNA was assumed to be independent of context. McClintock approached the genome with an intuition, intimacy, and a holistic view that included ecological context (see Keller, 1984). From these methods, she sensed and felt for the organism. Her intuition made all the difference. McClintock was awarded a Nobel prize in honor of her research demonstrating that genetic code is flexible, relational, and responsive to contexts.

There has been a long metaphorical slippage between DNA code and computer code. For both, code is widely understood as a logical process that cannot be questioned or modified, except by experts. For DNA code and computer code, Wendy Hui Kyong Chun argues that this metaphor of code creates a black box: we assume it works, and because it's working it must be correct (Chun, 2008). Blackboxing occurs any time complex technologies are used without questioning or understanding the design choices, thereby allowing the technology to be a mystery to the user. Chun identifies blackboxing throughout the history of computing, from the shift for early computers to use graphic interfaces, to the desktop metaphors of our current computers, to the open source movement. She posits that programming is "sold as offering the programmer more and easier control, but they also necessitated blackboxing even more the operations of the machines they supposedly instructed" (45).

Advocates of open source code may reply, "Github means that there is no blackboxing!" as if giving access to code is to offer complete transparency. This is precisely the promise of democratization of which Chun is most critical. She writes, "The history of computing is littered with moments of "computer liberation," that are also moments of greater obfuscation" (45). That obfuscation may be the relationship between hardware and software, the relationship between syntax and function, or the choices made by the programmer. The promise of transparency offered with open source code also obfuscates the complex work of code literacy, which is a complex social, embodied, and hegemonic process.

Most significantly for my feminist intervention, code itself hides the work of coding and the coder. When I make programming choices, I make them based on my experience, education, goals, values, and embodied process of composing. These elements of code literacy are overlooked with open source promises of transparency. The most famous example of this is likely the facial recognition software that was programmed to only recognize white faces. When this was first reported,[1] news headlines read "Facial Recognition Systems Show Rampant Racial Bias" (CNN) and "Facial-recognition Technology has a racial-bias problem" (Business Insider). These headlines place the technology in the subject position; the code is the actor. Hence, by this logic, the code has a racial-bias problem. Again, these headlines entirely erase the people who made decisions, the context in which they work, and the process of algorithmic writing. Does the technology itself have a racial-bias problem? Or the people who programmed?

For Chun, as well as myself, transparency is not the goal. After describing the layers of blackboxing in the history of computing, she "does not argue we need to move beyond specters and the undead, but rather contends that we should make our interfaces more productively spectral" (60). The promise of transparency itself is what she is skeptical of. There will always be unknowns, mysteries, layers we don't understand.

Depending on where you look, the history of computing can also be read as is also a history of intuition, emotion, and affection. Most famously, Ada Lovelace identified herself as the "high priestess of mathematics," and her letters show her deep devotion to both the logic of mathematics and the personal commitment to the metaphysical. Her letters to collaborator Charles Babbage are equal parts love letter and equal parts computer programming, and I'm not sure which got her blood pumping more, Charles or the mechanical brain.

Flash forward to 1953, we find Christopher Strachey, British mathematician and computer scientist, writing what was at the time the longest computer program. This program would also create the first computer generated writing. What, you may wonder, did Christopher teach this early computer to write? Love letters, of course. Strachey worked through the night on the Manchester Mark 1, or the baby as its operators called it. In the morning, the baby could write

[1] For a thorough and in-depth analysis of racist programming practices, *Algorithms of Oppression* by Safiya Noble.

awkward, clunky, endearing love letters with lines like “FANCIFUL MOPPET, MY WISTFUL LITTLE LIKING HOPES FOR YOUR ENCHANTMENT.” and “MY LOVEABLE HUNGER LONGS FOR YOUR LUST. YOURS IMPATIENTLY, M.U.C.” Be still my heart.

Just as Barbara McClintock proceeded with her research with a feeling for the organism, I proceeded with my own digital humanities methods, and, in particular, my work with learning to program in R with a *feeling for the algorithms*. This essay is a methodological narrative that outlines a heuristic for a feminist rhetoric of code. I base my heuristic on a collaborative research project published in *Peitho Journal*, with my mentors and collaborators Gesa Kirsch and Alison Williams. For that research, our methods included both close reading and distant reading, for which I learned to program to compose data visualizations. I will first offer an overview of that project and our goals. Then, I will identify what is at stake for my two primary audiences. First, I am writing for scholars studying algorithmic rhetoric and rhetoric of code.[2] For them I address the particular needs of this subfield for a deeper engagement with feminist methodology. Next, I am writing for feminist scholars for who I highlight emerging scholarship in feminist engagement with DH that I am indebted to for my own work.

I offer a heuristic for DH feminist methodology in 4 overlapping phases-- listening phase, dreaming phase, conversation phase, and tinkering phase.[3] In each phase, I will narrate how feminist rhetorical practices prepared me to learn and compose algorithmic rhetoric. I call it a heuristic, but I’m just telling stories. I’ve learned to value stories from the community of Cultural Rhetoric scholars and their methods of performing the deep ties between bodies, communities, and research (Powell, 2012; Powell et al., 2014; Cobos et al, 2018). In addition, feminists have drawn on stories in order to feature embodiment as central to theory (Johnson et al, 2015). I hope this article will inform future scholars as they envision digital humanities projects engaged with feminist rhetorical practices. I invite you, as well, to consider your own work with code, and the affective, social, lively aspects of the work.

Overview of Research Project on Women’s Medical Journal

Over the past two years, I’ve worked collaboratively with Gesa Kirsch and Alison Williams on the research presented in “Feminist Practices in Digital Humanities Research: Visualizing Women Physician’s Networks of Solidarity, Struggle and Exclusion,” recently published in *Peitho Journal*. Based on original archival material in the *Woman’s Medical Journal* from 1900-1919,

[2] Algorithmic rhetoric and rhetoric of code are aligned subfields within digital rhetorics. These subfields take up programming, code, and algorithms as rhetorical processes and rhetorical texts. This research enables scholars to both make rhetorical choices while composing in code as well as to identify and understand the rhetorical power embedded in computer programs.

[3] I note here that I am not outlining a pragmatic set of steps that others would necessarily want to follow. In fact, I encourage a more efficient process than mine, if you’re able. The narrative is intended to highlight my feminist process, which was embodied, affective, collaborative, and social.

we studied this community of early 20th century women physicians, their evolving community, rhetorics of solidarity, and the limits of that solidarity. In order to arrive at empathetic, nuanced claims, we had to view this community from a distance, up close, and also with a focus on the margins and erasures. For our feminist goals, these three methods and three ways of viewing informed each other so that we grappled with complex themes of community, solidarity, and exclusion while also placing this community in historical context and treating the women with empathy as well as accountability. This research required mixed methods, including close reading familiar to feminist scholars as well as innovative methods from digital humanities.

It is widely accepted that the process of interpretation and analysis involves intuition and subjective reading. However, many scholars of DH assume that subjectivity is limited to the interpretation. Beyond analysis, I feature the deeply intuitive process that was involved in every step of the methods including programming and writing with code. I outline this intuitive process in 4 main phases: listening phase, dreaming phase, conversation phase, and tinkering phase. In each of these phases, I collaborated with experienced programmers and programmed myself. In each phase, the programming opened new opportunities for feminist interventions and feminist rhetorical practices.

In this article, I am going to narrate how we created the visualizations like the one below:

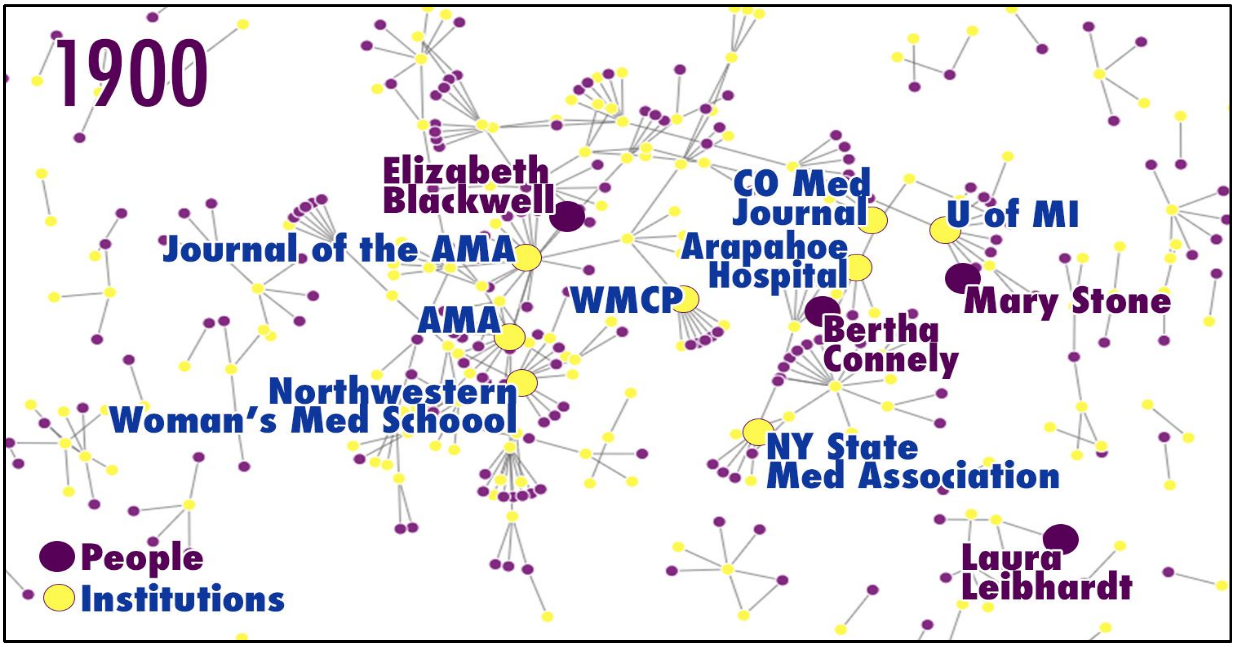

Figure 1: A social network analysis featuring people and institutions who were included in the 1900 archive of the *Women's Medical Journal*. This visualization was created in R[4] and labels added with Photoshop.

[4] R is an open source programming language that is most often used by social scientists. It is a robust language designed for large data sets and complex data relations. It's relatively easy to learn.

Due to the complex, layered mixed methods used in the project, there was very little space to explain the process for creating this visualization and no space to explain the programming choices we made. In fact, in the article reporting this research published in *Peitho*, we reduced the description of the method for creating the visualization to 1 sentence, effectively blackboxing nearly the entire process.

If I were to outline the method in a series of logical, efficient steps, I would offer these instructions:

> **Step 1:** Code the archival material for all "actors" defined as people and institutions and locations using a google form that creates a spreadsheet.
>
> **Step 2:** Upload spreadsheet to Google graphs and Tableaux, use network graph to create beta visualization.
>
> **Step 3:** Format and clean data.
>
> **Step 4:** Program in R using social network analysis and igraph packages to create visualizations, which included loading the packages igraph, networkD3, and htmltools.
>
> **Step 5:** Program in JavaScript using force-network to add interactivity.

Hypothetically, if you wanted to create a similar visualization, you could follow similar steps and thereby reproduce similar research. Emphasis on the *hypothetically.*

What We Do When We Program

In the step-by-step instructions above I've *excluded* 90% of the energy and intellectual labor. That 90% was the intuitive process whereby we arrived at these instructions. Herein lies the danger of black boxing algorithmic work for a feminist scholar: precisely what is removed is the embodiment, emotions, context and the feeling for the algorithm.

The rule-based, logical work with algorithms is preserved and celebrated. This is most obvious in previous work on coding rhetoric that centers on procedural rhetoric. Ian Bogost (2008) defines the work of programming as fundamentally procedural: "To write procedurally, one authors code that enforces rules to generate some kind of representation, rather than authoring the representation itself. Procedural systems generate behaviors based on rule-based models; they are machines capable of producing many outcomes, each conforming to the same overall guidelines" (122). Defining programming as authoring "code that enforces rules to generate a representation" is certainly accurate in a technical sense. However, the work looks, feels, and is enacted in a more complex way. Bogost essentially erases the labor and full process of writing with code because he is most interested in the play that code makes possible.

Behind this procedural rhetoric is the human programming, her goals, embodied experiences, histories, and attachments to the work. I would add, there's a fair amount of play involved throughout the process of programming. Annette Vee (2013) offers a detailed historical account of computational literacy,[5] placing programming as a literate practice that emerges within historical and cultural contexts. Understanding programming as literacy includes attention to the "powerful social and historical dynamics of composing code" (46). Vee defines computational literacy as "constellation of abilities to break a complex process down into small procedures and then express—or "write"—those procedures using the technology of code that may be "read" by a non-human entity such as a computer" (47). Given that my collaborators and I are first-and-foremost committed to feminist rhetorical practices, these feminist practices are central in my own constellation of abilities as I gained programming literacy.

I especially encourage scholars of rhetorical code studies and algorithmic rhetoric to consider the importance of feminist methodology and feminist rhetorical practices. The field of Computers and Composition broadly speaking has been led by feminists, informed by feminist research, and is a collaboration among feminist to the point that Estee Beck in conversation with Gail Hawisher and Cynthia Selfe (2013) discuss the field broadly as having "feminist leanings." However, emerging research on code studies and algorithmic rhetorics do not seem to be currently following this tradition. Lori BethDe Hertogh, Liz Lane, Jessica Ouellette tested these "feminist leanings" through a comprehensive, qualitative analysis of *Computers and Composition*. Their findings must taken seriously by all scholar algorithmic rhetoric committed to ethical and inclusive scholarship:

> "The period of 2009 to the present brought in an interesting mix of more recent publications on intersectionality and technofeminism, but at the same time, recent publication trends also indicate a shift (particularly in areas such as *rhetorical code studies*, *algorithmic rhetorics*, posthumanism, object-oriented ontology, and game theory) away from either explicit or implicit uses of feminist and intersectional frameworks." (10 emphasis mine)

This has also been my experience: there is a gap between the feminist leanings of computers and composition as a field and the particularly process-focused, a-political, seemingly gender neutral but also male-dominated authors in the emerging field of rhetorical code studies and algorithmic rhetorics. The shift to focus on code and algorithmic rhetoric has not fully included some of the shared values for scholarship in composition studies, including embodiment and intersectional frameworks.

Although scholars of rhetorical code have not integrated feminist and intersectional methodologies into their work, the goals of this emerging subfield do align with much of feminist scholarship, especially technofeminist scholarship. In *Rhetorical Code Studies,* Kevin Brock outlines the significance of composing in code for all composition scholars "the practice

[5] Parallel to Vee's history of programming literacy, one could build a gendered history of programming literacy. In this history, we could see that men and women often developed and given access to programming literacy in different spaces and with different foci. Importantly, in this literacy history, as women begin to increasing gain proficiency in a particular computational literacy, that literacy is then considered less specialized, "downsourced," paid less, and even defined as "not real" programming. This story is for another day.

of composing in code will absolutely benefit our subsequent rhetorical analysis and critique of code if we are to effect change in a world in which it is more or less impossible to escape the influence of software and digital technology" (151). Composing in code is one emerging yet exciting avenue for feminists to shape in our algorithmic landscape, or at least to gain the literacy to critically assess research methods that incorporate algorithmic methods. This work is well underway. I'll proceed with a brief review existing feminist scholarship that engages in digital humanities methods, especially composing in code.

Feminist Interventions in DH

In their 2013 article "Meaningful engagements: Feminist historiography and the digital humanities," Jessica Enoch and Jean Bessette invited feminist scholars to consider the fruitful potential of DH methods. Scholars of feminist rhetorics have widely adopted DH methods feminist historiography and especially for digital archival methods (Sullivan and Graban, 2010; Ramsey-Tobienne, 2012; Enoch, 2013; Enoch, Bessette, & VanHaitsma, 2014; Graban & Rose, 2014; Gutenson & Bachelor, 2016). There has been both enthusiasm for the new methods as well as a sense of caution, hesitant to take up new methods that do not obviously converge with feminist practices that foreground ethics of care, situated knowledge, and personal engagement.

Among digital humanities scholars, there is a growing body of research that explores the complex tensions and possibilities that arise when feminists adopt digital humanities methods (Bailey, 2011; Enoch & Bessette, 2013; Wernimont, 2013; Sayers, 2018) as well as research that makes visible the politics encoded into algorithms and computer assisted research methods (Kimme Hea, 2007; Grunwell, 2018; Edwards & Gelms, 2018; Blair, 2018; Beck, 2018; Noble, 2018). I am also indebted to the long tradition of Technofeminist research in the field of Computers and Composition, which is a field is committed to the embodied, political, and social dynamics that are encoded into technology (for reviews of this work see Haas et al, 2019; De Hertogh et al, 2019; Devoss, 2019; as well as my own modest attempts to contribute to this conversation Fancher, 2019). For me, technofeminism is empowering for its commitment to simultaneously engaging in critique as well as production, building, and creation with technology by and for feminists (Wajcman, 2004). This feminist scholarship, especially technofeminist scholarship, underscore that when we program, we bring our constellations of literacies and our embodied experiences. I offer a narrative of my own constellation of literacies as I engage in writing with code.

Feeling for the Code: Listening, Dreaming, Conversing, and Tinkering

First, there was a **listening phase**, where I read through the primary archival material in the *Woman's Medical Journal* and paid attention to what caught my eye or my ear, the announcements in particular are rich, celebratory, and confident. I waded through the complexity of their discourse. I listened to these women's advice and problems, and pondered their at times strange medical practices. I was impressed with the nuanced ways that they addressed issues like sex work, pay disparity, and sexist discrimination. I listened to my collaborators, what caught their ears, and their impressions. I enjoyed this phase of open listening.

Feminist methodology taught me to listen with care and empathy. In this listening phase, I needed to listen as Krista Ratcliffe has taught us: to let the discourse wash over, to seek understanding, and to attune myself to tropes of whiteness both in the archive as well as in my own analysis. From Jacqueline Royster and Gesa Kirsch, I've learned to listen with critical imagination, leading me to ask questions about what could have been, what may be missing, and to carefully attune myself to the affect, desires, dreams, and motivations of the women in the journal. This listening phase felt like time travel, immersing myself in another world and way of being. I listened towards understanding and towards empathy.

I also listened as I experimented with code. I listened by exploring and attuning myself to previous feminist and rhetorical studies DH project. I explored DH projects featured in the *Programming Historian* collection. The edited collections *Making Things and Drawing Boundaries* and *Bodies of Information: Intersectional Feminism and the Digital Humanities* include discussions of case studies, featured projects, and citations to a fascinating diversity of digital humanities projects that are also engaged with feminists, queer, and anti-racist critiques. In Rhetoric and Composition, the digital projects of Laurie Gries, Michael Faris and Derek Mueller have been especially inspiring and instructive. I listened to these projects, the rhetoric of the code, and how that rhetoric shapes their research. This listening included attuning myself to the values, frameworks, and paths that their research questions led them down.

Beck has argued that "algorithms are persuasive because of their performative nature and the values and beliefs embedded and encoded in their structures" (2016). These persuasive elements include inclusion and exclusion of data, systematic organization of information, "ideological beliefs of relationships." When I listened to the code, I also felt the suasive pull. I listened and attuned myself to programming options. From this listening, these research projects that integrate DH methods persuade their users to engage with specific modes of inquiry and relations between variables.

Even before this, the code persuades the researcher to think in programming terms. I had previously never thought of a woman physician as a 'variable,' but in order to create a visualization I needed to think in those terms. Thinking of women as variables made me uncomfortable, uneasy with the erasure of embodiment and situation. I had to sit with this discomfort and notice that my training as a feminist scholar has helped me value embodied experience. This was now in my bones. We couldn't simply let the visualizations carry the weight of the argument. By listening to this discomfort, we planned methods that integrate both distant and close reading.

In addition, code invites particular kinds of relationships and organizations. In the case of my visualization, there had to be a hierarchy. The line has to start at one point and go towards another. We experimented with different relationships: from institutions to people, people to institutions, also from person to person. Each of these offered different ways of reading and could each have led to a different research project.[6]

Much of this listening was also practicing: downloading tutorials, performing exercises in code, and fiddling with open source code.[7] Yes, I did learn how to program in this phase. More importantly, I learned what programming could do and how it may shape research questions and methods. When I listened to code as I also practiced programming, I felt myself get lost in the patterns, structure, and rhythm of the work. Programming has its own flow that is quite different from writing, in my experience. There is an interactive moving back and forth between code and visualization, code and test, code and instructions. The rhythm of the work was absorbing.

My training in feminist rhetorical practices enabled me to listen carefully throughout the research process. My attention to listening moved me to consider many different perspectives, approaches, and attune myself to a wide range of previous work. In addition, this listening led us to new questions, to think critically about assumptions, and to trust our intuition when and where we felt discomfort in the process. Hence, the feminist rhetorical practice of listening was a vital resource as we designed this project and as I shifted to participate in algorithmic rhetoric.

Then, there was the **dreaming phase**: This is the phase in which we imagined what could be possible. The dream started with Jacqueline Royster and Gesa Kirsch in their chapter "Social Circulation and Legacies of Mobility for Nineteenth Century Women" in *Rhetoric, Writing, and Circulation*. They imagine using digital humanities tools to visualize the movement of this community of women physicians. When Gesa, Alison, and I began collaborating, we took up Royster and Kirsch's questions and asked more. What would we gain by visualizing the community? What stories would these visualizations tell? What relationships or changes in relationships could we possibly locate?

We dreamed big. We played with mapping software, interactive storytelling, and timelines. I can't tell you how many hours I've spent scrolling through the D3 library of Javascript packages for data visualization. As I explored, I imagined how each algorithm may interact with the archival data. How could an algorithm help us see the community differently? What stories could another algorithm help us tell? When brainstorming, Gesa, Alison and I covered every inch of the walls of our work space with drawings, lists, sketches of the different ways we could imagine visualizing this community. We moved around the room drawing, arranging Post-Its of

[6] In fact, we created several of those different versions with different relationships, with a particular focus on relationships between people. However, that work ended up being beyond the scope of the arguments and stories we wanted to develop in our research.

[7] My experience of listening would have been far easier if I had benefited from Kevin Brock's *Rhetorical Code Studies*, especially the very rhetorically focused and also practical chapter "Composing in Code," which was published long after I moved past this phase.

various sizes and colors, and discussing our plans. We also moved outside, walking and running through long, warm evenings returning each day with coffee and fresh perspective on plans.

While we spent hours and days dreaming up possibilities, we also spent as much time whittling these dreams down to those that would best address our research questions. I still hold in my head and my notebooks the dreams of how many different ways algorithms can tell the stories of a community. This process of dreaming, imagining, questioning, and choosing was guided by our goals as feminist researchers. Again and again, we returned to issues of intersectionality and how the archive excluded Black Women. We needed to dream anew, imagining ways of resisting that erasure. When I share the code with fellow researchers, these dreams are concealed, hidden within the code. However, our dreams are precisely what guided our choices working with algorithmic rhetoric.

Then, there was a **conversation phase**: we talked to everyone. The UCSB digital humanities librarian, Thomas Padilla, pointed me towards resources, examples, and programs. I'm fortunate to be colleagues at UCSB with Jeremy Douglas, a DH scholar who has been generous and supportive. Gesa consulted friends and colleagues with expertise in programming, visualization as well as medical history. Alison was in close conversation with mentors. We shared their insights and resources, allowing our own social network to shape our research methods.

A major turning point in the project as well as my own literacy with digital humanities methods came when I attended Kairos Camp and benefited from the community of scholars, both instructors Cheryl Ball and Doug Eyman and the fellow students. And then the next year, Alison, Gesa, and I all attended, making great strides in the development and design of our research project. The conversations at Kairos Camp were instrumental for sussing out the right tools, coding methods, and processes that would help us answer some of our research questions. Most importantly, we were encouraged to focus first and foremost on the research question and our audience, not the research tool. Often, when learning a new tool, such as DH methods or algorithmic writing, it is easy to focus on what the tool can do, how it works, and the new affordances offered. With their many years of experience, Ball and Eyman mentored us to learn both the new DH methods as well while keeping in mind my goals as a researcher and the needs of our audiences.

I also engaged in frequent conversations with undergraduate student programmers. Working with the students is where the rubber hit the road. I addressed questions like how I planned to turn archival material into programmable data, how I defined variables, how much data was enough or too much, and what could be the relationships between those variables. The students moved me to speak in precise, concrete plans that they could use to aid in coding the archival material and prototyping visualizations. In our conversations, I felt a closeness build, as we oscillated between the roles of teacher and student. As we took our turns teaching and learning, these relationships grew into partnerships through collaboration.

I was also in constant conversation with my collaborators, Gesa Kirsch and Alison Williams. This project emerged within and through our dialogue together. In conversation with them, we thought about the visualizations with a keen attention to feminist research practices: How can we preserve an ethic of care while representing this community? How can distant reading feature situated knowledge and account for context? What assumptions are we making? Who may be marginalized or excluded through our methodological choices? How may our own situated, embodied experiences be shaping our methods, the visualizations, and our analyses? I brought the voices and care of these intelligent, feminist collaborators with me as I programmed.

Throughout this conversation phase, I noticed myself gaining confidence and fluency in debates around digital humanities, methodologies, and ethics of those methods. It was through dialogue and collaboration that I grew as a researcher while also refining the methods. Through this phase, I am indebted the long-documented practices of feminist collaboration (Ede & Lunsford, 2001). In addition, feminist scholars in Rhetoric and Composition have narrated their own broad, multidirectional mentoring networks (Gaillet, Aley & Horner, 1994; Bloom, 2007; Gaillet & Guglielmo, 2014; VanHaitsma & Ceraso, 2017). In our research project and among feminist scholars generally, mentoring and collaborating are intertwining practices. Be they student, colleague, friend or partner, each person who shared their time and experience served as a vital mentor as I grew into the identity of digital humanities researcher and as we refined our research methods.

Then, there was the longest phase, which was the **tinkering phase**. In this phase I toggled between coding the archival materials, testing out a program, going back and re-coding, trying out new programs. Giving up on especially cumbersome programs. Asking for help…a lot.

I call this phase “tinkering” because I want to highlight a process of negotiation with computer programs and programming languages. Tinkering highlights the kind of relationship I have with algorithms: I can change, adjust, fiddle with the program, but I cannot build it from scratch and I do not have total control over the final product. The algorithmic logic itself maintains some control, recalcitrant against my fiddling at times and flexible at other times. At the same time, the algorithm does not control everything.

Jacqueline Wernimont’s feminist approach to DH also resonates with my experience of tinkering: “‘Interaction’ resonates with the ongoing emphasis on collaboration in the field, but also suggests the use or inhabiting of the space between actions – between ‘use’ and ‘creation/making,’ or between ‘making’ and ‘theorizing.’” She defines feminist DH work at the threshold between actions. I’ve also found that to be the case. The algorithm and I interacted: I acted and it reacted, and vice versa. I’m at the threshold between coder, archiver, researcher, feminist, teacher, collaborator, and programmer. I interact with each of these thresholds.

While tinkering, I created the prototypes in programs Google graphs and Tableaux. While neither of these programs gave me the flexibility and control that we needed, they allowed me to make prototypes of visualizations. With each prototype, we could see the community in slightly different ways. The prototyping process helped us to decide what kind of visualizations would help us tell this story.

For example, the image below was made in Google graphs. This was a prototype that helped us decide the direction of the final visualization. Our final visualization looks quite different and tells a more complex story. However, they both tell an important part of the story: the American Medical Association was a powerful organization in these women physicians' professional lives, even though the AMA did not widely accept women as members.

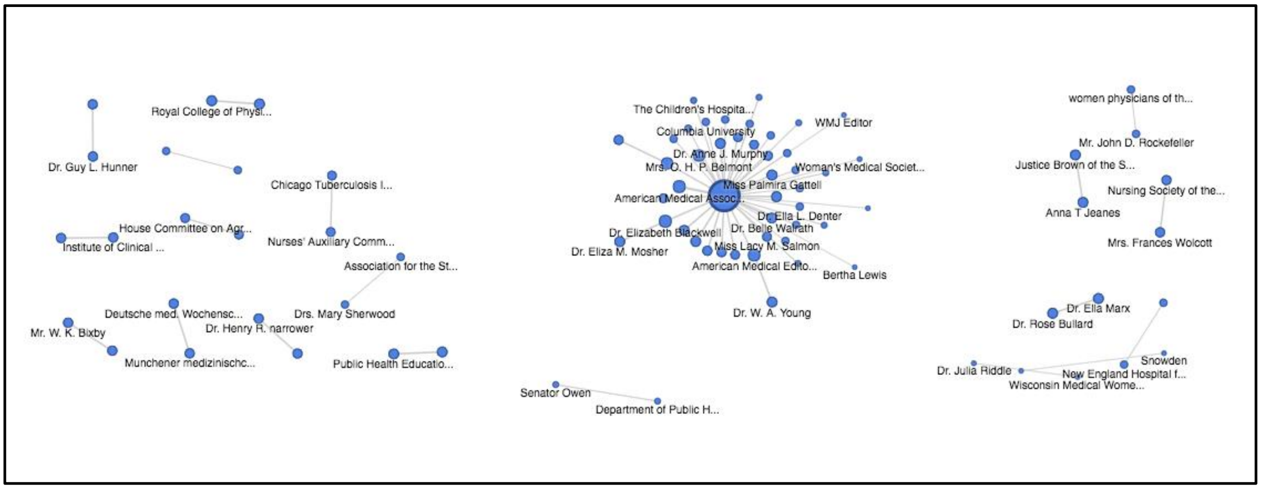
Royal College of Physi...
Dr. Guy L. Hunner
Chicago Tuberculosis I...
House Committee on Agr...
Institute of Clinical ...
Nurses' Auxiliary Comm...
Association for the St...
Deutsche med. Wochensc...
Drs. Mary Sherwood
Dr. Henry R. narrower
Mr. W. K. Bixby
Munchener medizinischc...
Public Health Educatio...
Senator Owen
Department of Public H...
The Children's Hospita...
WMJ Editor
Columbia University
Dr. Anne J. Murphy
Mrs. O. H. P. Belmont
Woman's Medical Societ...
Miss Palmira Gattell
American Medical Assoc...
Dr. Ella L. Denter
Dr. Elizabeth Blackwell
Dr. Belle Walrath
Dr. Eliza M. Mosher
Miss Lacy M. Salmon
American Medical Edito...
Bertha Lewis
Dr. W. A. Young
women physicians of th...
Mr. John D. Rockefeller
Justice Brown of the S...
Nursing Society of the...
Anna T Jeanes
Mrs. Frances Wolcott
Dr. Ella Marx
Dr. Rose Bullard
Snowden
Dr. Julia Riddle
New England Hospital f...
Wisconsin Medical Wome...

Figure 2: An early prototype of the social network analysis featuring people and institutions who were included in the 1900 archive of the *Women's Medical Journal*. This visualization was created in Google Graph.

While this prototyping process was not programming, it was still working with algorithms as a means of research because the visualizations were generated through Google graphs or Tableaux's algorithms. At this point, I had little control over the algorithm. However, working with these programs allowed me to see the community and the archival material in different ways. With Gesa and Alison, we thought with the algorithms insofar as we imagined how these visualizations could best help us to study this community of women.

Most of the decisions made in this phase were based on one question: does this tell an interesting story? Based on how we answered this question, we would go back to the drawing board: Do we need more or less data? Why did we include or exclude some texts? What variables will we include? How do we format our data? How can we cut down on errors?

When I first learned R, I was frustrated. There were days when it seemed like nothing would work. A series of red error messages seemed to be my fate. At the most basic level, there are a few main steps:

- download a package,
- create a dataframe,
- define a table with values,
- and assign a function.

Those steps appear simple. And in some ways, it is easy to play with simple data sets and graphs to get started. The frustration comes in the fact that there are many different ways to design each step, with different values, variables, design choices, and relations between data. The frustration also arose when I created visualizations, and they didn't appear as I had imagined or planned. This phase taught me to practice patience and care. In particular, I had to be patient with myself, letting myself learn and grow would take time.

After a while, there were days when I felt like I found my flow. You could walk by my office door and hear Pranati Shah, a research assistant I worked with, and me cheering and giving each other high fives. On Friday afternoons, Pranati would teach me something new, but we both learned along the way through a process of trial and error. After a while, the errors became fewer and the trials led to usable visualizations. My shoulders relaxed. We lost track of time, tinkering in my office side by side on warm fall afternoons. By the end of our time working together, each Friday I looked forward to experimenting, trying out new data, and eating pizza with Pranati.

Finally, I tinkered with the design of the visualization, which included a shift to think more actively about aesthetics, audience, and usability. What colors or interactive options should we include? What determines the center? margins? size? proximity? There is no particular correct answer to these questions, but there were better and worse choices. For instance, the student research assistant I worked with initially designed the social network analysis with pink dots for people and blue dots for institutions, a color choice so traditionally gendered that it would have inflected interpretations that we were not comfortable with.

This tinkering phase is when I really developed a feeling for the algorithm. At the same time, I brought to this work each of the earlier phases. I continued to listen, but this time I was listening while programming. I continued to dream, but these dreams focused on the specific variables and possibilities afforded in the programming languages. Of course, conversation and dialogue with Gesa Kirsch and Alison Williams guided each interaction with the algorithm. In addition, they were my first audience. Their responses and reactions became my first round of user testing. Together, we discussed design and usability options. Throughout every step, this feeling for the algorithm emerged through my embodied, social, and feminist practices.

Tinkering was a vital feminist practice for my own algorithmic methods because tinkering foregrounds interaction and interative process. I didn't wield mastery over the algorithm. Nor was I entirely subject to its demands. Instead, I was able to recognize my own power to enact changes as well as the limits of that power. I was already familiar with this notion of change by learning from the generations of feminist activists that came before me, including the women physicians in the archival materials. Feminist activism is far more significant than the term 'tinkering' suggests. But in this instance, I felt empowered to tinker with programming and learn a new literacy because of the broader constellation of literacies, including feminist rhetorics, which prepared me for collaborative, interative, and experimental process of composing.

Conclusion

I hope my readers note that, in each of these phases, my work emerged out of my embodied experience, my communities, feminist values, and social contexts. I've narrated my process in order to highlight what algorithmic composition looked like for me. I am no stereotype of a programmer, which I imagine as white guy with poor posture, pale skin, working alone listening to EDM. That's not me, and that's not how I've worked. Most importantly, I did not work alone nor did I try to exclude my values from my work. Rather, I worked actively to think as a feminist who was working with algorithmic writing towards feminist rhetorical goals. In the black box of the instruction set, I define only a few of the steps that I took while programming. However, in order to arrive at these steps, I drew upon a community and am indebted to that community. I've also drawn upon my embodied experiences and values, which shaped my process and the product of my programming.

I stress the community because one of the earliest arguments for the parallel between composition and coding (Cummings, 2006) revises the rhetorical triangle. Rather than the traditional author, audience and text, Robert E. Cummings defines the coding rhetorical triangle as coder, machine, program. When we adopt coding as a means of composition, yes, the machine does become one audience, but it is not the only audience. We still program for and with people. Our audiences are many. And the algorithmic rhetoric that we compose will interact with all bodies in different ways. In addition, I am not just composing with code. I am not just a programmer; I'm also an author, with my goals, values, biases, and priorities. I am a teacher, friend, feminist and colleague. My students, friends, and colleagues are vital resources for my programming literacy.

I am a feminist, and I code as such. I find it vitally important to resist a shift to over-determine programming through its relationship to the machine, thereby isolating the programmer and machine apart from the broader social context.

I also call attention to my social and embodied experiences as a novice programmer in order to invite us to think differently about the process of working with algorithmic rhetorics. Moya Z Bailey invites DH scholars "challenge the "add and stir" model of diversity" because "as opposed to meeting people where they are, where people of color, women, people with disabilities are already engaged in digital projects, there's a making of room at an already established table." If we are to meet feminist scholars where they are in Digital Humanities research, we would also be inviting a new set of questions, processes, and experiences. In this essay, I've highlighted the embodied, collaborative, and the social process of programming in order to center these foundational concepts of feminist writing and rhetoric within the compositional process of writing with code.

I seek to resist the presupposition that feminist scholars are deficient or lacking in skills necessary to complete DH research. Jessica Enoch and Jean Bessette, while they invite feminist interventions in digital humanities research, also note that technical skills may be a limitation or a barrier to feminist scholars. Other feminist scholars may feel the same way. To them I say: our expertise and experience as feminist researchers are assets, rich resources that qualify and prepare us to complete nuanced, ethical, important research in digital humanities. Let these feminist practices shape the technical work.

Acknowledgements

First and foremost, I must express my deep gratitude to Gesa Kirsch and Allison Williams, my collaborators, mentors, friends. While I led the technical aspects of this project, they supported, guided, problem-solved, and cheered me along the way. With the generous support of the CCCC Emerging Research Grant, I was able to have time to dedicate to this project and funding for student researchers. I am indebted to Cheryl Ball and the staff of KairosCamp for their intellectual generosity and material support for this project. Finally, where would I be without my brilliant students? I'm certain that the work of Raul Eulogio, Pranati Shah, and Ari Gilmore not only made this project possible but also mentored me as I gained literacy and competency. Finally, Jason Helms is a mentor and friend in so many ways, and his support and insights shaped the final version of this essay. Thank you, Jason, for listening to my stories and helping them shine.

References

Bailey, M. Z. (2011). All the digital humanists are white, all the nerds are men, but some of us are brave. *Journal of Digital Humanities 1*(1). http://journalofdigitalhumanities.org/1-1/all-the-digital-humanists-are-white-all-the-nerds-are-men-but-some-of-us-are-brave-by-moya-z-bailey/

Beck, E. (2013). Reflecting upon the past, sitting with the present, and charting our future: Gail Hawisher and Cynthia Selfe discussing the community of computers & composition. *Computers and Composition*, *30*(4), 349-357.

Beck, E. (2016). A theory of persuasive computer algorithms for rhetorical code studies. *Enculturation*, *23*. http://enculturation.net/a-theory-of-persuasive-computer-algorithms

Beck, E. (2018). Implications of persuasive computer algorithms. In J. Alexander & J. Rhodes (Eds.), *Routledge companion to digital writing & rhetoric* (pp. 291-302). Routledge.

Blair, Kristine. (2018). Technofeminist storiographies: Talking back to gendered rhetorics of technology. In J. Alexander & J. Rhodes (Eds.), *Routledge companion to digital writing & rhetoric* (pp. 224-241). Routledge.

Bloom, L. Z. (2007). Mentoring as mosaic: Life as guerilla theater. *Composition Studies*, *35*(2), 87-99.

Bogost, I. (2008). The rhetoric of video games. In K. Salen (Ed.), *The ecology of games: Connecting youth, games, and learning* (pp. 117-140). MIT Press.

Brock, K. (2019). *Rhetorical code studies: Discovering arguments in and around code*. University of Michigan Press.

Chun, W. H. K. (2008). *Control and freedom: Power and paranoia in the age of fiber optics*. MIT Press.

Cobos, C., Raquel Ríos, G., Johnson Sackey, D., Sano-Franchini, J., & Haas, A. M. (2018). Interfacing cultural rhetorics: A history and a call. *Rhetoric Review*, *37*(2), 139-154.

Cummings, R. E. (2006). Coding with power: Toward a rhetoric of computer coding and composition. *Computers and Composition*, *23*(4), 430-443.

De Hertogh, L. B., Lane, L., & Ouellette, J. (2019). "Feminist Leanings": Tracing technofeminist and intersectional practices and values in three decades of computers and composition. *Computers and Composition*, *51*, 4-13.

DeVoss, D. (2019). TechnoFeminisms: A conversation about pasts, presents, and futures. *Computers and Composition*, *51*, 68-78.

Ede, L., & Lunsford, A. A. (2001). Collaboration and concepts of authorship. *PMLA*, *116*(2), 354-369.

Edwards D.W. & Gelm B. (2018) The rhetorics of platforms: Definitions, approaches, futures. *Present Tense*. *6*(3). http://www.presenttensejournal.org/editorial/vol-6-3-special-issue-on-the-rhetoric-of-platforms/

Enoch, J., & Bessette, J. (2013). Meaningful engagements: Feminist historiography and the digital humanities. *College Composition and Communication*, *64*(4), 634–660.

Enoch, J. (2013). Coalition talk: Feminist historiography: What's the digital humanities got to do with it?. *Peitho Journal*, *15*(2), 40-45. https://cfshrc.org/article/coalition-talk-feminist-historiography-whats-the-digital-humanities-got-to-do-with-it/

Enoch, J., Bessette, J., & VanHaitsma, P. (2014). Feminist invitations to digital historiography. *Sweetland Digital Rhetoric Collective.* Retrieved from https://www.digitalrhetoriccollaborative.org/2014/03/28/feminist-invitations-to-digital-historiography/

Eyman, D., & Ball, C. E. (2014). Composing for digital publication: Rhetoric, design, code. *Composition Studies*, *42*(1), 114-117.

Fancher, Patricia. (2019). TechnoFeminist design. TechnoFeminism:(Re)generations and intersectional futures. *Computers and Composition Online*, *51*. http://cconlinejournal.org/techfem_si/04_Fancher/

Fancher, Patricia, Gesa Kirsch, & Alison Williams. (2020). Feminist practices in Digital humanities research: Visualizing women physician's networks of solidarity, struggle and exclusion. *Peitho Journal*, *22*(1). https://cfshrc.org/article/feminist-practices-in-digital-humanities-research-visualizing-women-physicians-networks-of-solidarity-struggle-and-exclusion/

Johnson, M., Levy, D., Manthey, K., & Novotny, M. (2015). Embodiment: Embodying feminist rhetorics. *Peitho Journal*, *18*(1), 39-44. https://cfshrc.org/article/embodiment-embodying-feminist-rhetorics/

Keller, E.F. (1984) *A feeling for the organism: the life and work of Barbara McClintock*. Macmillan.

Graban, Tarez, Richard Urban, and Stephen McLeroy. *Linked women pedagogues project*. http://lwpproject.org/wp/

Graban, T. S., & Rose, S. K. (2014). Editors' introduction: The critical place of the networked archive. *Peitho Journal*, *2*(1). https://cfshrc.org/journal/peitho-volume-17-issue-1-fall-winter-2014/

Gries, L. (2017). Mapping Obama hope: A data visualization project for visual rhetorics. *Kairos: A Journal of Rhetoric, Technology, and Pedagogy 21*(2). https://kairos.technorhetoric.net/21.2/topoi/gries/index.html#

Grunwell L. (2018). Constructing research, constructing the platform: Algorithms and the rhetoricity of social media research. *Present Tense. 6*(3). https://www.presenttensejournal.org/volume-6/constructing-research-constructing-the-platform-algorithms-and-the-rhetoricity-of-social-media-research/

Gutenson, L. D., & Bachelor, M. (2016). Race, women, methods, and access: A journey through cyberspace and back. *Peitho Journal*, *19*(1). https://cfshrc.org/article/race-women-methods-and-access-a-journey-through-cyberspace-and-back/

Haas, A., Rhodes, J., & DeVoss, D. N. (2019). Introduction to a special issue on TechnoFeminisms. *Computers and Composition*, *51*, 4-13.

Kimme Hea, A. C. (2007). Riding the wave: Articulating a critical methodology for web research practices. In H. A. McKee & D. N. DeVoss (Eds.), *Digital writing research: Technologies, methodologies, and ethical issues* (pp. 269-286). Hampton Press.

Losh, E., & Wernimont, J. (Eds.). (2019). *Bodies of Information: Intersectional feminism and the digital humanities.* University of Minnesota Press.

Mueller, D. N. (2017). *Network sense: Methods for visualizing a discipline*. WAC Clearinghouse.

Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. NYU Press.

Powell, M. (2012). 2012 CCCC Chair's Address: Stories take place: A performance in one act. *College Composition and Communication*, 64(2), 383-406.

Powell, M., Levy, D., Riley-Mukavetz, A., Brooks-Gillies, M., Novotny, M., & Fisch-Ferguson, J. (2014). Our story begins here: Constellating cultural rhetorics. *Enculturation*, *25*. http://enculturation.net/our-story-begins-here

Ramsey-Tobienne, A. E. (2012). Archives 2.0: Digital archives and the formation of new research methods. *Peitho Journal*, *15*(1), 4-29. https://cfshrc.org/article/archives-2-0-digital-archives-and-the-formation-of-new-research-methods-2/

Ratcliffe, K. (2005). *Rhetorical listening: Identification, gender, whiteness.* SIU Press.

Royster, J. J., & Kirsch, G. (2018). Social circulation and legacies of mobility for nineteenth century women. In L. Gries, & C. G. Brooke (Eds.), *Circulation, writing, and rhetoric* (pp. 170-188). University Press of Colorado.

Sayers, J. (Ed.). (2018). *Making things and drawing boundaries: Experiments in the digital humanities*. University of Minnesota Press.

Sullivan, P., & Graban, T. S. (2010). Digital and dustfree: A conversation on the possibilities of digital-only searching for third-wave historical recovery. *Peitho Journal*, *13*(2), 2-11.

Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies*, *1*(2), 42-64.

Wernimont, J. (2013). Whence feminism? Assessing feminist interventions in digital literary archives. *DHQ: Digital Humanities Quarterly*, *7*(1). http://www.digitalhumanities.org/dhq/vol/7/1/000156/000156.html

Wajcman, J. (2004). *TechnoFeminism*. Polity.