

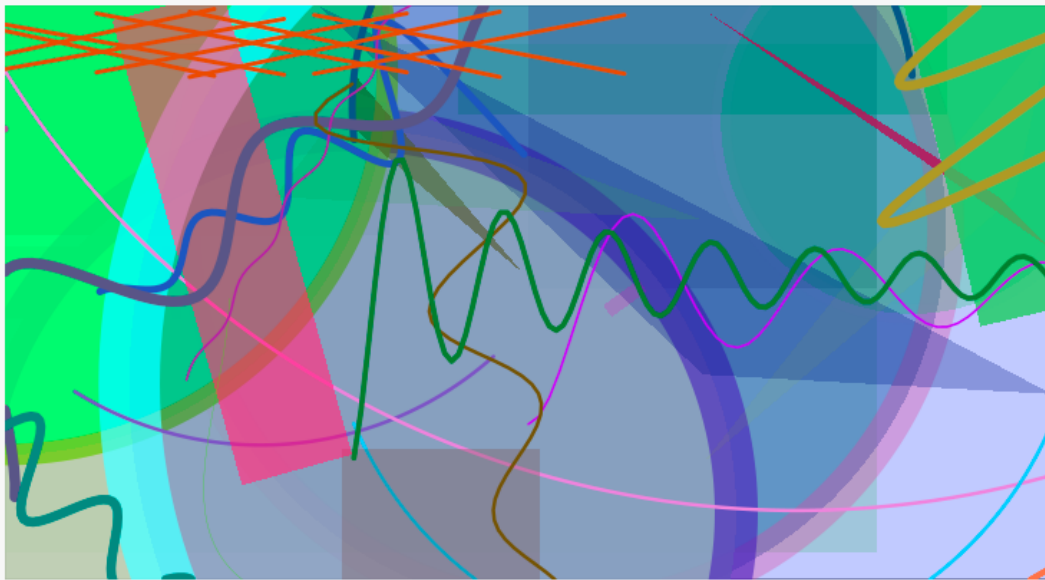
## Precision art

### Background

On the [impute.me](#) website one of the modules is named “[kandinskify yourself](#)”. It has been running almost since the founding on the project in 2015 and is based on [this algorithm](#). It’s purpose is to generate a simple artwork based on randomness - but with the checksum of the users DNA as random seed. In this way the art is unique to the user themselves and will be so even on re-running of the proceessing. It has been accessed by approximately 10k users, which is ~25% of the total user base. A typical output looks like this:

Making unique art from your genomic data is not a new idea. However, most places I've seen that offer such service actually just use very few SNPs. In this module all your trait-associated SNPs are combined to create a truly unique piece of art in the style of [Wassily Kandinsky](#), using the beautiful code available in the [kandinsky](#) R-package.

You may ask if the drawing reveals deeper insight about your genome and inner self? Maybe it does, maybe it doesn't. Either way - it is guaranteed to be a unique drawing derived only from your genome. You may print it and use it as art. I do.



*A screenshot of the output of the current kandinskify algorithm*

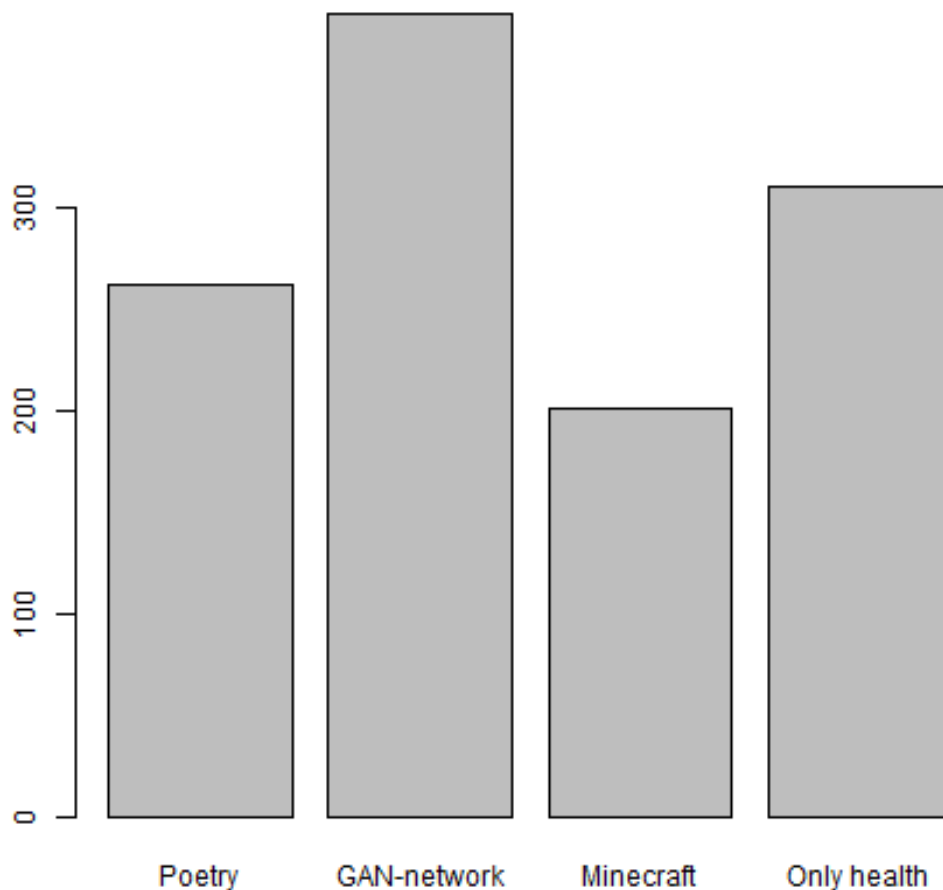
Since 2019 an additional plugin have performed the same, but with written poetry based on [this algorithm](#), such as this example.

Realized dashing rose from palm tree leaf to harrowing wind,  
hidden doors drawn by many channels, a  
neurotic ripple begins to seize.  
But I should be untrue to genetics, scratching among its inevitable methylation sites  
so let us attempt to tell a story without pathway redundancies.  
An algorithm reading your genome, 2020-02-06

The technical implementation of both these is fairly simple ([script](#)) and is only based on the property of DNA-information as a random seed. This can be improved. For this reason we launched a user survey in summer 2019 posing the following four choices

1. Focus on the written poetry: explore how the AI-generated poetry can be made even more relevant and personal, based on trait and ancestry predictions
2. Generating GAN-network-based art spanning from Rococo landscapes to modern art, then training user-impressions on genetics to provide incrementally better and better personalized precision-art
3. Generating downloadable Minecraft worlds based on your personal DNA, complete with walkable double helix tunnels, gene-annotations and warp-points for travelling to your favourite gene
4. Dont spend time on this, stick to health genetics!

The results of the survey is now available and came out in favour of option 2:



It is therefore of interest to expand on this into a module that will be called the *precision art* module.

## Suggested implementation

The main criteria of success is to showcase modern image machine learning algorithms and how they can function with genetics. In other words - if it is interesting, it is appropriate. A little more detailed setup idea consist of two separate work packages. A work-package 1 which expands the scope of pictures generated using [GAN](#)-approaches and a work-package 2 which trains the outputting generator based on genetic information and user-feedback such that it can generate images that are increasingly liked by people, as deducted from their DNA information.

For work-package 1, many interesting approaches are outlined online, e.g. [this](#), [this](#), [this](#), or [this](#). All have the capability to act as image-generators based on provided data sets. No doubt more are available. The key to create here is to figure out how to connect the generated image with genetic data. Presumably a checksum of DNA data could suffice, such that people with a given DNA code always get a unique image.

For work-package 2, the task involves setting up (in R/Shiny) a simple interface that can collect user-input given on each image as e.g. thumbs up or thumbs down. With that in hand, as well as the complete set of derived DNA-calculations from impute.me, it should be possible to train the image-selector such that it can optimize the count of thumbs up scores. If this is achieved the module will become the worlds first *precision art* module. Within the scope of work-package 2, test data running is enough, leaving room...

For work-package 3: which simply is the real-life implementation in impute.me including collection of user feedback and continous automated improvement of image-selection algorithms.

The sole criteria for evaluating success of this project is the same as with any art project: ability to catch interest of viewers. As such, these work-packages should only be seen as suggested guidelines.

## Resources available

The following resources will be made available.

1. **Hardware:** SSH-based access to an AWS cloud computer of the g3s.xlarge size, 100 GB storage, 1 GPU processor with 8 GB memory, and 4 CPU processors with 30 GB memory.
2. **Standard software:** Ubuntu (18.04), with R (3.4.4) and python (2.7.17), CUDA Toolkit (10.2, for GPU usage). But the instance user, is super-user, so can install whatever else is needed using *sudo*.
3. **Genomics data:** In the ~/data folder genomic data from 51 example-genomes are found in standard impute-me format: The *id\_613z86871.input\_data.zip* file contains all directly measured genotypes. The *simple\_format* and *gen* format files contain more extensive imputed information. The json format file contains all derived genetic

calculations and may be an alternative source for genetic information. The remaining files are fast-access subsets or non-important information. All are better [described here](#).

4. **Art repository data:** Large image repositories are available online, and finding the right one may be key to this project. However, two large sets are already pre-loaded on the computer: The [CelebA](#) data set of 200k aligned portrait images and a web-scrape from wikiart of 10k various art images, with more details can be found on how to perform similar manual web-scrapes on different genres using this [script](#).
5. **Custom software:** Each computer has a pre-configured copy of the impute.me web-analysis server running on shiny. This is for (optional) implementation in work-package 3. The ~/srv folder contains the running code ([from here](#)), the ~/programs contains required support software, e.g. impute2 and shapeit. ~/imputations is an empty folder for the queuing system where uploaded data will appear. You can browse the graphical interface using any internet browser. Enter your host IP address + :3838 which is the standard shiny port (so `http://ec2-35-164-169-143.us-west-2.compute.amazonaws.com:3838`)

## Login and setup information

1. Each team get one instance each.
2. Pick a random one from this [google-doc list](#). You may leave a comment to say it's taken.
3. Log in using the info (if in doubt how, then use free [mobaxterm](#)).
4. Immediately change password using command `sudo passwd ubuntu`. Choose password only your team knows.
5. If you can't log in, pick another instance
6. If there are none left, write [lasse.folkersen@regionh.dk](mailto:lasse.folkersen@regionh.dk) then we start some more.
7. If you completely mess up your instance, write [lasse.folkersen@regionh.dk](mailto:lasse.folkersen@regionh.dk) and get a new one (-1 point for style).