**NIPT-PG: Empowering Non-Invasive Prenatal Testing to learn from population genomics through an incremental pan-genomic approach**

Zhengfa Xue**1,2,†**, Aifen Zhou**3,4,†**, Xiaoyan Zhu**1,2**, Linxuan Li**5,6**, Huanhuan Zhu**5**, Xin Jin**5,7,\***, Jiayin Wang**1,2,\***

**1** School of Computer Science and Technology, Xi’an Jiaotong University, Xi’an, China.

**2** Shaanxi Engineering Research Center of Medical and Health Big Data, Xi’an Jiaotong University, Xi’an, China.

**3** Institute of Maternal and Child Health, Wuhan Children’s Hospital (Wuhan Maternal and Child Health care Hospital), Tongji Medical College, Huazhong University of Science and Technology, Wuhan, China.

**4** Department of Obstetrics, Wuhan Children’s Hospital (Wuhan Maternal and Child Health care Hospital), Tongji Medical College, Huazhong University of Science and Technology, Wuhan, China.

**5** BGI Research, Shenzhen, China.

**6** College of Life Sciences, University of Chinese Academy of Sciences, Beijing, China.

**7** School of Medicine, South China University of Technology, Guangzhou 510006, China.

**†**These authors have contributed equally to this work and share first authorship.

\*Correspondence: Jiayin Wang, wangjiayin@mail.xjtu.edu.cn

Xin Jin, jinxin@genomics.cn

**Supplementary Materials**

**1. The usage of NIPT-PG**

NIPT-PG is a software specifically designed for the detection of chromosomal aneuploidies in extremely low-coverage sequencing data in non-invasive prenatal testing (NIPT). It primarily focuses on common fetal chromosomal disorders such as Down syndrome (Trisomy 21), Edwards syndrome (Trisomy 18), and Patau syndrome (Trisomy 13). The software comprises the following steps: I. Generate a pan-genome from a specified number of sequencing files (.sam). II. Account for alignment errors for each read during the alignment process, realigning sample reads to the pan-genome. III. Calculate the distribution range of sequencing depth for multi-source aligned reads in the new sample and employ an optimized z-score for detecting aneuploidy. For specific details, refer to the Materials and Methods section in the main text. Instructions for usage are as follows:

**Step 1. Package Dependency**

first, install the NIPT-PG conda environment:

*conda create -c NIPT -PG*

*conda activate NIPT-PG*

then, in NIPT-PG environment, install the following package:

*pip install pandas numpy tqdm argparse*

**Step 2. generating pan-genome**

**usage:**

*python3 gen\_pgg.py [-r REF.FA\_FILE] [-s SAM\_PATH] [-n NIPT\_FILE]*

**optional arguments:**

*-r* path to the reference genome file (such as GRCh38.fa)

*-s* path to the folder containing the files to be tested

*-n* path to the nipt\_files.csv

**example:**

*python3 gen\_pgg.py -r data/ref.fa -s data/sam/ -n data/nipt\_files\_ART-Random.csv*

The content of the nipt\_files.csv file is as illustrated in **Table 1**, documenting the file name mappings for each testing file. This practice aids in standardizing file management and enhances testing efficiency.

**Table 1**. Illustration of the nipt\_files.csv file.

|  |  |  |
| --- | --- | --- |
|  | nipt\_files | mapping |
| 0 | CL100050702\_L02\_91 | sample\_0 |
| 1 | CL100025607\_L02\_22 | sample\_1 |
| 2 | CL100035831\_L01\_15 | sample\_3 |
| … | … | … |

**Step 3. Sequence-to-graph alignment**

**usage:**

*python3 map2pgg.py [-p PGG\_FILE] [-s SAM\_PATH] [-n NIPT\_FILE] [-k K\_MER]*

**optional arguments:**

*-p* the pan-genome file path

*-s* path to the folder containing the files to be tested

*-n* path to the nipt\_files.csv

*-k* k-mer length, default=5

**example:**

*python3 map2pgg.py -p data/pgg.json -s data/sam/ -n data/nipt\_files\_ART-Random.csv -k 5*

**Step 4. Z-score test based on multi-source aligned read**

**usage:**

*python3 aneup\_det.py [-s SAM\_PATH] [-g ALIGNED\_SAM\_PATH] [-n NIPT\_FILE]*

*[-l LEFT\_THRESHOLD]*

*[-r RIGHT\_THRESHOLD]*

*[-c* *CONTROL SAMPLE]*

**optional arguments:**

-*s* path to the folder containing the files to be tested

-*g* path to the folder containing realigned samples

-*n* path to the nipt\_files.csv

-*l* left threshold of z-score (default = -3)

-*r* right threshold of z-score (default = 3)

**example:**

*python3 aneup\_det.py -s data/sam/ -g data/aligned\_sam/ -n data/nipt\_files\_ART-Random.csv -l -3 -r 3*

**2. Description of the simulated sequencing data**

NIPT queues typically originate from specific geographic regions. Shared variant loci within the population are identified by NIPT-PG as polymorphic rather than mismatch loci, representing a key advantage of NIPT-PG. Therefore, we generated two types of simulated data, namely, population samples and randomly mutated samples, to assess the performance of NIPT-PG. The main software tools utilized include MS [1], SEQ-GEN [2], and ART [3]. The program MS can be employed to generate numerous independent replicate samples under various assumptions regarding migration, recombination rates, and population size, aiding in the exploration of polymorphism studies. SQE-GEN is capable of simulating the evolution of sequences under various mutation models. ART constitutes a suite of simulation tools designed to generate next-generation sequencing data from a given reference genome.

**3. Population samples**

Firstly, we simulated the evolutionary trees using the following command in the MS software:

*ms 2000 24 -T | tail -n +4 | grep -v // > treefile*

Here, we generated 24 chromosomes, each with 2000 population samples. When the option -T is used the trees representing the history of the sampled chromosomes are output. For example, the command line *ms 5 2 -T* results in the following output:

//

((2:0.074,5:0.074):0.296,(1:0.311,(3:0.123,4:0.123):0.187):0.060);

//

(2:1.766,(4:0.505,(3:0.222,(1:0.163,5:0.163):0.059):0.283):1.261);

This output represents the trees for two samples. The tree format is the Newick format utilized by Phylip and a number of other applications. The branch lengths are in units of 4*N0* generations, where N0 is the diploid population size. The sampled chromosomes are labelled 1, 2 ... corresponding to ordered sampled chromosomes. The tail +4 command is used to strip off the first four lines output by ms. The grep -v // command removes the lines with double slashes from the ms output. Subsequently, SEQ-GEN was utilized to generate reference genomes for trisomy samples based on the evolutionary trees:

*seq-gen -mHKY -l 100000 -s .2 <treefile1128 >seqfile*

The -mHKY command for seq-gen specifies (in this case) the JukesCantor mutation model. The -s command for seq-gen scales the branch lengths so that θ per site is 0.2, in this case. Here, we set the length of each chromosome to be 100,000. The content of the seqfile is as follows:

//

2000 100000

1 GGCGTCCGGCCAAAGGTTCTTGACACACGATACCTTAGTT…

2 GGCGTCCGGCCAAAGGTTCTTGACACACGATACCTTAGTT…

3 GGCGACCTGACAAAGGGTCTTGACACACGATACGTTAGTT…

4 GGCGACCGGACAAAGGGTCTTGACACACGATACGTTAGTT…

5 GGCGACCGGACAAAGGGTCTTGACACACGATACGTTAGTT…

…

//

2000 100000

1 ACCAGGGCTCGGAACTCTCTGGGTTTGCCTAAGTGCCCTT…

2 ACCAGGGCTCGGATCTCTCTGGGTTTGCCTAAGTGCCCTT…

3 ACCAGGGCTCGGATCTCTCTGGGTTTGCCTAAGTGCCCTT…

4 ACCAGAGCTCGCTACTATCCGGGTGTGCCTAAGATCCCAC…

5 ACCAGAGCTCGCTACTTTACGGGTGTGCCTTAGTGGCGCA…

…

Individual samples within the population share a largely common genome, with only a small portion of shared variations. Finally, we extracted the reference genomes of individual samples from the seqfile, and then simulated the sequencing files of individual samples using the ART software. The command used is as follows:

*art\_illumina -ss HS25 -sam -i ./ref\_MS+Seq-Gen+ART\_sample1.fa -l 35 -p -f $random\_float -m 100 -s 5 -M -ir 0.005 -ir2 0.005 -dr 0.005 -dr2 0.005 -na -o "./sam-ms/${output\_file}"*

Here, "art\_illumina" denotes the use of the Illumina sequencing platform with the HS25 model. The read length is 35 base pairs (bp), which is consistent with the read length of real NIPT sequencing data. The insertion rate for both the first-read and second-read is 0.005, and the deletion rate for both reads is 0.005. The mean size of DNA/RNA fragments for paired-end simulations is 100, with a standard deviation of 10 for the DNA/RNA fragment size in paired-end simulations. The term "random\_float" represents random coverage between 0.1 and 0.5. Its command is as follows:

*random\_float=$(awk -v min=0.1 -v max=0.5 'BEGIN{srand(); print min+rand()\*(max-min)}')*

The two thousand samples were divided into two parts. One thousand samples underwent no processing, while the remaining one thousand samples were proportionally divided into three parts and subjected to trisomy perturbations. Initially, we extracted chromosomes 13, 18, and 21 from the reference genome as new reference genomes, denoted as ref13, ref18, and ref21. Subsequently, using the ART software, we generated one thousand sequencing samples based on a single chromosome in proportion, with a sequencing depth of 0.1 and other parameters as mentioned above. Its command is as follows:

*$art -ss HS25 -sam -i ./ref\_MS+Seq-Gen+ART\_sample1\_chr13.fa -l 35 -p -f 0.2 -m 100 -s 5 -M -ir 0.005 -ir2 0.005 -dr 0.005 -dr2 0.005 -na -o "./sam-ms/trisomy-ms/${output\_file}"*

In addition, for each chromosome, we randomly added ±0.005x coverage to simulate the normal fluctuations in the data. In the final step, we removed the header data from these trisomy perturbation sequencing samples based on a single chromosome and integrated them with the original 1000 samples. Due to the influence of maternal DNA, the chromosomal amplification factor for trisomic chromosomes in the fetus is slightly reduced. Therefore, here we designated trisomy perturbation samples with original coverage between 0.1 and 0.25 as trisomic samples, with approximately 1.4-2 times amplification for chromosomes 13, 18, and 21 in these samples.

**4. Randomly mutated samples**

For randomly mutated samples, we directly employed the ART software for simulation. Its command is as follows:

*$art -ss HS25 -sam -i ./ref\_ART-Random.fa -l 35 -p -f $random\_float -m 100 -s 5 -M -ir 0.005 -ir2 0.005 -dr 0.005 -dr2 0.005 -na -o "./sam/${output\_file}"*

Unlike the population samples, random mutation samples use a uniform reference genome with a total length of 2.4 million base pairs. All other parameters for ART simulation are exactly the same as those used for population sample simulation.

When simulating sequencing data with the ART software, a small number of sequencing errors are generated. After removing these samples, the population samples (*MS+Seq-Gen+ART*) and the randomly mutated samples (*ART-Random*) retained 1859 and 1847 samples, respectively. Among the population samples, there are 156 trisomy 13 cases, 131 trisomy 18 cases, and 148 trisomy 21 cases. For the randomly mutated samples, there are 151 trisomy 13 cases, 140 trisomy 18 cases, and 148 trisomy 21 cases.

**5. NIPT data with different fetal concentrations**

In order to investigate the detection accuracy of NIPT-PG at different fetal concentrations, we simulated multiple sets of NIPT data based on *ART* and *MS*, varying fetal concentrations. We utilized MS template data identical to the population data (without regenerating the pan-genome), with sequencing coverage set randomly between 0.1x and 0.5x. We simulated a total of 6000 NIPT samples, with 1000 serving as normal baseline samples. For each chromosome, we randomly added ±0.005x coverage to simulate the normal fluctuations in the data, without any additional modifications. Next, 5000 samples were designated as trisomy 13 samples. These samples were divided into ten groups, each comprising 500 samples. Fetal concentrations were individually set at 4%, 6%, 8%, 10%, 12%, 14%, 16%, 18%, 20%, and 22%, respectively. Furthermore, in addition to simulating the normal fluctuations in data coverage, these 5000 samples underwent trisomy processing. Assuming a sample has N reads on chr13, in the group with fetal concentration F, the number of fetal reads on chr13 would be *N × F*. After trisomy processing, there would be a total of *N + d × N × F* reads (where 0.5 ≤ *d* ≤ 1) on chr13. To achieve this, an augmentation of the original reads counts in the sample by *d × N × F* reads is required. The method of increasing reads is similar to the one mentioned earlier. Ultimately, we obtained non-euploid sample data at different fetal concentrations.

**6. The impact of pan-genome graph scale on NIPT-PG**

We simulated two sets of samples (population data and random mutation data) using MS and ART software, with each set comprising 3000 samples. The reference genome used for both sets is consistent with the supplementary materials 3 and 4. After excluding samples with sequencing errors, we retained 2000 samples from each set. For each dataset, we progressively built pan-genomes using 200, 400, 600, ..., 1800, 2000 samples, which were then employed for the detection of population data and random mutation data in Table 1 of the main text. The pan-genome construction followed a stepwise accumulation approach, where the initial selection of 200 samples formed the first pan-genome, and subsequent pan-genome sizes increased in increments of 200 samples. Ultimately, for both sets of samples, we obtained ten pan-genomes of varying sizes to assess the impact of different pan-genome scales on NIPT-PG.

**Reference**

[1] Hudson RR. Generating samples under a Wright–Fisher neutral model of genetic variation. Bioinformatics 2002;18(2):337-8.

[2] Rambaut A, Grass NC. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Bioinformatics 1997;13(3):235-8.

[3] Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. Bioinformatics 2012;28(4):593-4.