TRUFFLE improvements section

Apostolos Dimitromanolakis

## Planned Improvements to TRUFFLE

The current version of truffle has been developed with a focus in speed and ease of use,  and is readily applicable to small to medium-size datasets. For example it is able to analyse the full 2504 individual  from the 1000 Genomes dataset in less than 5 minutes on an Intel i7 desktop computer. However, applying truffle to population level datasets would benefit greatly if the analysis speed was increased by 10 to 50 times. The current implementation is not optimized to the full extent possible and would benefit from incorporating additional ideas, algorithms and technical tweaks to achieve the maximum speed possible.

Hashing and dictionary based methods have been used for haplotype matching e.g. (Durbin 2014; Gusev et al. 2009) and provide faster lookups of genotypes using a dictionary based approach. This can greatly improve the speed of segment matching algorithms, although at a cost of higher probability of missed segments because of genotyping error. Theoretically this method can be used for segment identification, in linear time, with out having to search through all the pairs of individuals in a study. In practise, the complexities of handling genotyping error reduces the speed benefit somewhat. We plan to integrate a hybrid approach, where a simplified model would be used to detect individual pairs that might share a segment and then running the full model of truffle, which reliably account for genotyping error,  on those pairs.

Locality-sensitive hashing (LSH) (Wang et al. 2013; Har-Peled, Indyk, and Motwani 2012) is a relatively new approach that allows one to quickly find similar entries in large collections of sequences. This approach belongs to a novel and interesting class of algorithms that apply projections of points to reduce the dimensionality of data while at the same time maintaining the approximate distances between the points. Although this idea has not been applied to the segment detection setting, we believe that this approach holds significant promise for improving matching speed and enable handling of population level datasets with ease. Applying LSH would improve on genotyping error tolerance of hashing methods such as (Gusev et al. 2009).

Technical improvements can also provide significant gains in execution speed. By using specialised processor instructions, one can perform arithmetic operations on vector data in a single step, by the use of vectorized instructions (Cockshott and Renfrew 2004; Kusswurm 2014) (e.g SSE3 and AVX) which areavailable on all modern computer processors. These instructions have been widely applied to the popular genomic analysis software PLINK (Purcell et al. 2007)  and have provided great improvements in speed. We plan to develop specific functions to utilise such processor features when available and estimate that an improvement in speed of 2 to 8 times is possible.

Another important area of improvement is enabling the more user-friendly analysis of multi-million individual datasets. Currently truffle, like most relatedness analysis software, has a single file as input. However, this might not be practical when population level datasets grow in size. In such cases, a likely scenario would be that subsets of individuals are split into different files, which would make computing relationships between all pairs an exercise in file joining and splitting. We plan to incorporate features that enable the automatic analysis of such split datasets. In addition, large sequencing studies sometimes can result in single sample VCF files, which would require a joining step before segment sharing analysis. To enable a more straightforward pipeline we plan to add handling of such files in truffle.

In addition, we plan to improve the applicability of truffle in a cluster computing environment. This would typically be done by enabling the automatic assignment of subsets of individuals to different cluster nodes. Previously we have implemented such functionality in the software package PREST-plus (Sun and Dimitromanolakis 2014; Sun and Dimitromanolakis 2012). We plan to extend truffle and enable similar functionality to provide a very easily applicable solution for running truffle in a computer cluster.

**Accuracy improvements**

Currently the error model in truffle derived to be tolerant to genotyping errors that occur in genotyping array based datasets. We have tested truffle in sequencing data and have demonstrated the applicability to high coverage sequencing data. When analyzing low-coverage data sequencing data, the current error model would likely not be adequate. We plan to extend the error model and validate it in high genotyping error cases of 0.5% to 2%. This is typically the achilles heel of most segment detection algorithms, as they cannot cope easily with high genotyping error, especially Hidden Markov Models and hash based approaches.

In addition, we plan to improve the handling of missing data within truffle, and allow easily handling of hard to genotype areas of the genome where missing rate in the resulting genotypes would be high.

Relatedness and inheritance of genomic segments is typically measure only on autosomal data by most software packages. The relatedness estimates in X-chromosome can be wildly different from the autosomes, as it follows a different inheritance pattern. We plan to implement integrated X-chromosome analysis in truffle to make it one of the few software that can provide relationship estimation using both autosomal and X-chromosome data. Because of the lower recombination rate it would typically require different models for the analysis and discovery of shared segments. We plan to include the 3 different models that would incorporate the sex status of the individuals in consideration. We plan to extend the current model for the analysis of female/female pairs and create two new models for analysis of male/male and male/female combinations. The pseudo-autosomal region of X would also require specific handling for the analysis purposes.

Efficient haplotype matching and storage using the positional Burrows–Wheeler transform (PBWT) Richard Durbin

 current state

improving speed

ways to facilitate large scale dataset analysis

improving for exome sequencing data

## Meeting points for inclusion

PAGE LENGTH OF TEXT THAT describe potential speed up and general improvements

- methods to improve truffle for application to 100000k individuals

- speed up

 - currently we do this, what can we do

-  hold back some speedups we have already done for grant

-  ways to split the load to  multiple cluster instances

- error rate in highly repetitive regions

    regions of genome with high segment sharing

  LD pruning and improvements for marker correlation

  exome sequencing data    (x)  (x)

+ text on how accuracy  will be determined in a rigorous way

MUMITA  X chromosome data (future)

# References

Durbin, R. 2014. “Efficient Haplotype Matching and Storage Using the Positional Burrows-Wheeler Transform (PBWT).”. *Bioinformatics* 30: 1266–72.

Gusev, A, JK Lowe, M Stoffel, MJ Daly, D Altshuler, JL Breslow, JM Friedman, and I Pe’er. 2009. “Whole Population, Genome-Wide Mapping of Hidden Relatedness.”. *Genome Res* 19: 318–26.

Wang, Hongya, Jiao Cao, LihChyun Shu, and Davood Rafiei. 2013. “Locality Sensitive Hashing Revisited”. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management - CIKM 13*. ACM Press. doi:10.1145/2505515.2505765.

Har-Peled, Sariel, Piotr Indyk, and Rajeev Motwani. 2012. “”. *Theory of Computing* 8 (1). Theory of Computing Exchange: 321–50. doi:10.4086/toc.2012.v008a014.

Cockshott, Paul, and Kenneth Renfrew. 2004. “Intel SIMD Instructions”. In *SIMD Programming Manual for Linux and Windows*, 47–97. Springer London. doi:10.1007/978-1-4471-3862-4\_4.

Kusswurm, Daniel. 2014. “X86-AVX Programming - Packed Integers”. In *Modern X86 Assembly Language Programming*, 405–37. Apress. doi:10.1007/978-1-4842-0064-3\_15.

Purcell, Shaun, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel A.R. Ferreira, David Bender, Julian Maller, et al. 2007. “PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses”. *The American Journal of Human Genetics* 81 (3). Elsevier BV: 559–75. doi:10.1086/519795.

Sun, Lei, and Apostolos Dimitromanolakis. 2014. “PREST-plus Identifies Pedigree Errors and Cryptic Relatedness in the GAW18 Sample Using Genome-Wide SNP Data”. *BMC Proceedings* 8 (Suppl 1). Springer Nature: S23. doi:10.1186/1753-6561-8-s1-s23.

Sun, L, and A Dimitromanolakis. 2012. “Identifying Cryptic Relationships.”. *Methods Mol Biol* 850: 47–57.