University of Bergen
Department of Mathematics

Inferring CRCs progression dynamic with HyperTraPS

Author: Hsiu-Jane Chen

Supervisors: Iain G. Johnston

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

06.July 2021

Abstract

This study investigated the applicability of novel HyperTraPS (hypercubic transition path sampling) platform for inferring the likely dynamic pathways of feature acquisition in colorectal cancer progresses. Cross-sectional data of genetic traits (chromosomal aberrations in this study) was applied a Bayesian approach and the multiple competing pathways for feature acquisition underlying given observation was inferred by the posterior of model presenting as directed acyclic graph (DAG). The novelty of allowing each genetic trait owning multiple orderings in the HyperTraPS models enable the dispersal of feature acquisition across state space to be displayed; multiple competing pathways of feature acquisition underlying given observation could alone be inferred by the posterior. Graph on progression dynamic of feature acquisition and the probabilistic graph inferred by the posterior output of model, confirm the power of model to predict out-of-sample observations. Results of model confirm the mainly findings of other studies, they suggest also other potential exploration of data. The flexibility and ability of capable handle high dimensional data promise as a suitable model for elaborated study of progression dynamic of colorectal cancer in the future.

Acknowledgements

I would first like to thank my supervisor Iain G. Johnston for introducing me to statistical methods that I had not learned before and making sure that I was on track. Thanks to his insightful feedback pushed me to sharpen my thinking and patient support to provided me with the tools that I needed to choose the right direction. Finally, I could not have completed this thesis without the support of my friends, Chung-Wei Weng, who given me advice and helped me to deal with technical problems. Hsiu-Jane Chen

06 July, 2021

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Colorectal cancer is a cancer that occurs in the colon or rectum. It is the third most common cancer on the world. According to the colorectal cancer rate ranking, Norway ranks on the fourth place of the world. Like many other biomedical problem, the occurrence of colorectal cancer involves in serial of stochastic acquisition of genetic traits aberrations. Understanding the dynamics of the carcinogenesis process has the potential to predict future biological behavior of tumor progresses and to learn effective therapies and [Greenbury et al., 2020].

1.1 progression dynamic

Here we assume the biological process as a process of evolution of biological traits involve in. During the process, trait moves around possible states of space and takes possible orderings relating to other traits associated to. Trait change can be an independent process or correlated with the presence of other traits involving in the same process. In the biological processes, some evolution of trait need quite long time and others in short term, such like the mutation of genes in virus. The progression dynamic that we are interested in is an order of trait change, but not the absolute temporal dimension of trait change.

1.1.1 Models of progression dynamic

Model based approaches which adopt computational methods to find the progression dynamic of genetic traits of biological processes in recent 20 years were reviewed in [Beerenwinkel et al., 2015, Schwartz and Schäffer, 2017]. The occurrence of cancer is supposed to involve in serial of stochastic acquisition of genetic traits aberrations. Traits used for approaches addressed different levels of genetic alterations, from cancer associated copy-number alterations (CNAs) at chromosomal arms, mutations of functional pathways to mutations of single gene. [Vogelstein et al., 1988, Fearon and Vogelstein, 1990] presented a linear sequential model of temporal order of CNAs acquisition consisting of 4 chromosomal aberrations, initialing with 5q- and following by 12p-, 18q- and 17p-. This model was derived from the comparative relative occurrence frequency of mutations of specific genes, whose alterations were assumed to be responsible for tumorigenesis, in tumor probe tissues of 4 different stages of tumor progression.

1.1.2 Oncogenetic tree model

This seen for others for simplified model was short after replaced by oncogenetic tree model [Desper et al., 1999, Szabo and Boucher, 2002]. Model assumed the causal relationship between genetic traits is tree like and the causalities between different pairs of genetic tree are independent. The presence of genetic trait is supposed to be random and distributed as Poisson process. The probability of presence or absence of pair of traits is defined as the edge probability of model.

Model begins at the root of tree where no trait presents and the next trait is sampled only if the difference between the calculated edge probability between the two sequential edges larger than a chosen threshold. The weight function is defined as the log-likelihood of all potential combinations of model derived from supposed criteria. Because the sampling process propagates towards the maximization of model weight, model with the maximal branching is chosen as the resulting model of pathway dynamic of traits. Cancer progression is then reconstructed as independent acyclic paths with branches and not allowing the convergence of the paths.

However, this kind of models because of demanding computational calculation allow only few genetic traits involving in. [Höglund et al., 2002] criticised that the potential existence of overlapping between different groups of genetic alterations was not taken into account in model that he found out their existence in his study.

1.1.3 Conjunctive Bayesian Networks

Alternatively, [Gerstung et al., 2009] applied 11 CNAs features to hidden Conjunctive Bayesian Networks (H-CBN) model to explore the dynamic of feature acquisition. H-CBN, abbreviation of Hidden Conjunctive Bayesian Networks that it is assumed that the mechanism of CBN dynamic is hidden by the observation of chromosomal aberrations applied to model. In CBN model [Beerenwinkel et al., 2007], the state in the transition network is a set of combination of the presence and absence of associated genetic traits. The model is then the family of probability distribution of various states ordered after the rule of order ideal. At the ground state 0, it is assumed that no event, also no genetic trait, exists. The distribution origins at the ground state and moves towards states locating at higher orders with increasing number of events. States with the same number of events are located at the same order of model.

The occurrence probability of a traits in a state is namely the ratio of trait in the state and in states at lower order of model. Hence, the probability of observing a state is namely the product of the summed probability of occurrence of traits appeared in state and the probability of not occurred ot traits that are associated to system but absent in state. Parameters in model were estimated through maximal likelihood estimation. Likelihood of model is the summation of probability of specific combination of states and the best model is this with the highest likelihood. The resulting model was presented in a directed acyclic graph (DAG). Model relax the dependence constraint of choosing the following state on the previous state and allows any arbitrary partial order. However, the relationship between states is monotonic and the influence of the presence of one state on the one next on is directed and not reciprocal. In addition, number of genetic traits applied to model is limited, otherwise the computation time increase exponentially with the number of traits.

1.1.4 Markov chain model

In [Hjelm et al., 2006], a stochastic models employing Markov chain has been applied to CHG data for acquisitions of colorectal cancer. In this model, the state is no more a single trait of chromosomal aberration, but a group of traits of chromosomal aberration, called module. Such design is purpose to avoid the problem of over-fitting in case of applying big size of traits to modelling.
Simulating the progress dynamic of traits as a Markov chain, it allows usually not so many traits applied to that the number of parameters needed scales exponentially. We need for each pair of traits a parameter that for n traits applied to model, we need 2^n parameters. Hence, in this model, traits in the same module are restricted to have the same dependence that number of parameters we need for a module with n traits is then n^2. Model allows only pairwise dependence among modules and the best model was derived from the model with the highest likelihood.
The likelihood of model is the summation of all probability of transition steps among modules. The calculation of likelihood can become intractable if the number of traits is too large. Likelihood in the model is the summation of probability of transition steps among modules of all possible scenarios. Supposed n traits was chosen for study, for a given dataset D with k of the n traits, the time for likelihood computation of D is $O(n2^k$ using of dynamic programming. For L observation data, we need altogether $O(nL2^k$. The computation of likelihood becomes intractable if n raise up til over 50.
The intractable computation problem with big size of traits , as it usually is in biological process, applied to model, is the central problem to construct the progress dynamic model for biological processes.

1.1.5 HyperTraPS

A recent approach, HyperTraPS (hypercubic transition path sampling) "presents progressive dynamics as paths on hypercubic space connecting all possible paths of traits presence and absence" [Greenbury et al., 2020]. Model addresses the exiting problem of limited number of coupled traits and has been applied a dataset with 65 observational genetic traits to find the dynamic pathways which addressed specific evolutionary questions

[Johnston and Williams, 2016].
Now exists a general platform for its application, interpretation, and visualization been constructed. HyperTraPS is embedded in a platform for parametric inference and model selection. Platform allows also Bayesian inference of dynamic pathways and identification of the model structures that best describe the dynamics and interactions contained within the observational dataset [Greenbury et al., 2020]. In response to the argument addressed by [Diaz-Uriarte, 2018] that features in cancer progresses may have multiple orderings duing to the high-dimensional structure of fitness landscapes and the potential presence of epistatic effects between genetic traits, the HyperTraPS platform directly allows this inference of multiple paths [Greenbury et al., 2020]. In the part of theory and method, I will describe detailed about important characteristics of models.

1.2 Statistics based clinical research

In the cancer research field, some disease-related genetic alterations are identified as hallmarks of cancer progression [Hanahan and Weinberg, 2000, Hanahan and Weinberg, 2011]. Exploring tumors samples from various stages of carcinogenesis, researchers attempted to clarify the the dynamic of genetic alterations in the cancer progresses. Traits used for approaches addressed different levels of genetic alterations, from cancer associated copy-number alterations (CNAs) at chromosomal arms, mutations of functional pathways to mutations of single gene.
The copy-number alterations (CNAs) at chromosomal arms was observed by applying the comparative genome hybridization (CGH) technology, in which the ”gains”, denoted as ”+” and the ”losses”, denoted as ”-” of features on the long arm (denoted as p) or short arm (denoted as q) of one of the 23 pair of human chromosomes are recorded. Such like the existence of group of CNAs, 8q+,13q+, 20q+, 8p-, 15q-, 17p-, 18q-, were identified important for the progression from adenoma to carcinoma [Ried et al., 2019]. [Sheffer et al., 2009] found the CNAs associated to different stages of tumors on basis of CGH finding on tumors samples of different stages and samples with or without impair of specific functional pathways. He concluded the existence of 4p-, 8p-, 15q- and 18q- an indicator of poor diagnosis of cancer progresses and of 4p-, 8p- and 15p- the bad survival chance for patients of colorectal cancer. [Hermsen et al., 2002] has, in addition to present the different cancer stages

associated CNAs, explored the correlation between the existence of various groups of CNAs. He concluded the positive and negative correlations between the existence of CNAs within and between groups. In spite of inconsistent agreements on the temporal order of genetic alterations in cancer progresses, some features, such like 20q+ was confirmed as one of the initial CNAs. Some chromosomal aberrations are to date widely used in clinical decisions as biomarkers for prognosis to treatment on colorectal cancer [Ried et al., 2019].

In this study, I applied observational data of chromosomal aberrations from colorectal tumors to the HyperTraPS platform to learn the temporal order of feature acquisition in cancer progresses. I took a Bayesian approach, using MCMC to estimate a posterior edge weights distribution over possible dynamic pathways across the hypercubic. The inferred pathways of cancer progresses were visualized through DAG. I discussed the findings in this study relating to knowledge on chromosomal aberrations associated to colorectal cancer progresses in existing in clinical or computer driven statistical studies. I undertook a comparative study on the most likely temporal order of feature acquisition inferred by HyperTraPS and this by the H-CBN in [Gerstung et al., 2009].

Chapter 2

Theory and method

This chapter introduces research material used in this study and the HyperTraPS model[Greenbury et al., 20 used for sampling and inferring transitional pathway of biological traits. The working pipeline of this study shows in figure 2.1

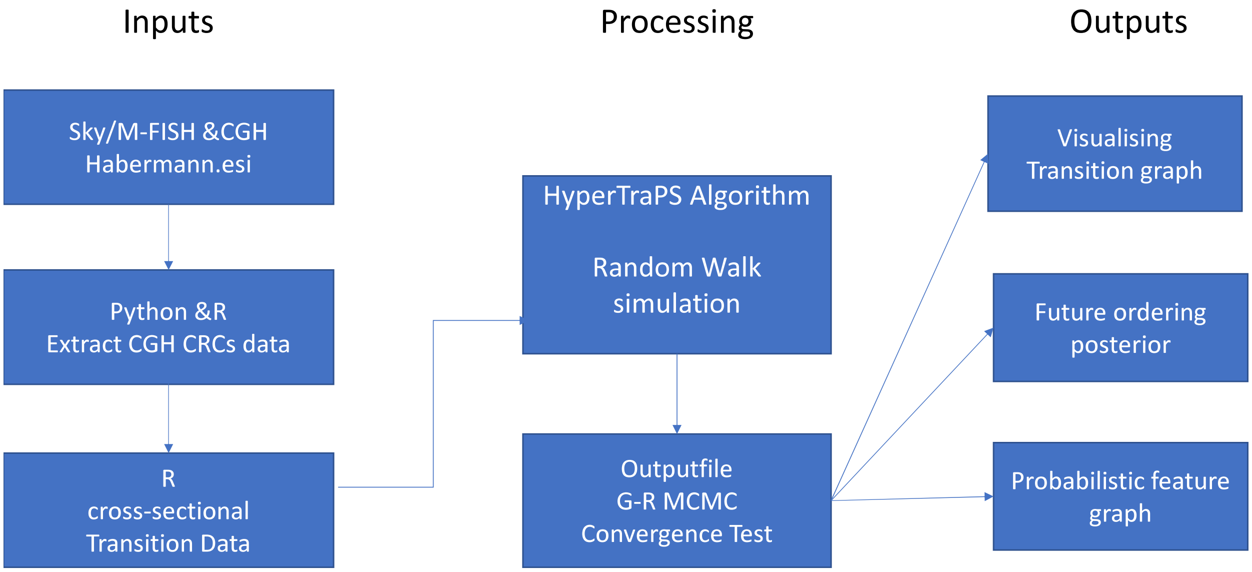
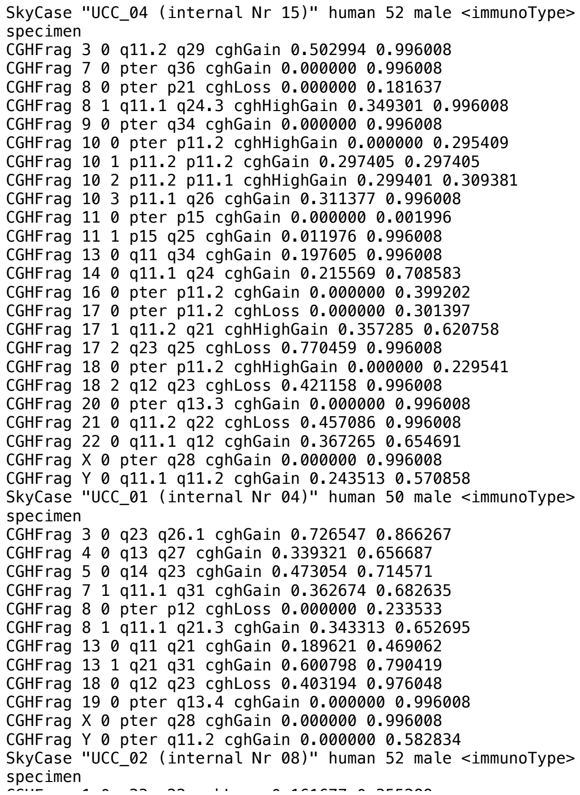


Figure 2.1: Habermann source data

2.1 Material

This study used the observations of chromosomes with copy number aberrations (CNAs) obtained by applying Comparative Genomic Hybridization (CGH) to tumors of patients of colorectal cancer. CGH is a molecular cytogenetic method for analysing copy number variations (CNVs) relative to ploidy level in the DNA of a test sample compared to a reference sample. Technique compare two genomic DNA samples which are often closely related to explore the differences, with regard to either gains or losses of either whole chromosomes or subchromosomal regions. For each one of the 23 pairs of human chromosomes (chromosome pair 1-22, chromosome X, chromosome Y),the long arm is denoted ”p” and the short ”q”. Data is downloaded from the platform SKY/M-FISH, a platform to which allow all investigators to share molecular cytogenic data[Knutsen et al., 2005] from their studies. The downloaded file is submitted by Jens K. Habermann, J.Habermann.esi, and contains samples from the 124 tumors of 19 patients. File contains not only information about the CNAs of chromosomal arms, but also others. I have written Python and R scripts using on it to create suitable file format to employ to HyperTraPS. This is a fraction of this file after extracting the part recording chromosomal aberrations.



```
SkyCase "UCC_04 (internal Nr 15)" human 52 male <immunoType>
specimen
CGHFrag 3 0 q11.2 q29 cghGain 0.502994 0.996008
CGHFrag 7 0 pter q36 cghGain 0.000000 0.996008
CGHFrag 8 0 pter p21 cghLoss 0.000000 0.181637
CGHFrag 8 1 q11.1 q24.3 cghHighGain 0.349301 0.996008
CGHFrag 9 0 pter q34 cghGain 0.000000 0.996008
CGHFrag 10 0 pter p11.2 cghHighGain 0.000000 0.295409
CGHFrag 10 1 p11.2 p11.2 cghGain 0.297405 0.297405
CGHFrag 10 2 p11.2 p11.1 cghHighGain 0.299401 0.309381
CGHFrag 10 3 p11.1 q26 cghGain 0.311377 0.996008
CGHFrag 11 0 pter p15 cghGain 0.000000 0.001996
CGHFrag 11 1 p15 q25 cghGain 0.011976 0.996008
CGHFrag 13 0 q11 q34 cghGain 0.197605 0.996008
CGHFrag 14 0 q11.1 q24 cghGain 0.215569 0.708583
CGHFrag 16 0 pter p11.2 cghGain 0.000000 0.399202
CGHFrag 17 0 pter p11.2 cghLoss 0.000000 0.301397
CGHFrag 17 1 q11.2 q21 cghHighGain 0.357285 0.620758
CGHFrag 17 2 q23 q25 cghLoss 0.770459 0.996008
CGHFrag 18 0 pter p11.2 cghHighGain 0.000000 0.229541
CGHFrag 18 2 q12 q23 cghLoss 0.421158 0.996008
CGHFrag 20 0 pter q13.3 cghGain 0.000000 0.996008
CGHFrag 21 0 q11.2 q22 cghLoss 0.457086 0.996008
CGHFrag 22 0 q11.1 q12 cghGain 0.367265 0.654691
CGHFrag X 0 pter q28 cghGain 0.000000 0.996008
CGHFrag Y 0 q11.1 q11.2 cghGain 0.243513 0.570858
SkyCase "UCC_01 (internal Nr 04)" human 50 male <immunoType>
specimen
CGHFrag 3 0 q23 q26.1 cghGain 0.726547 0.866267
CGHFrag 4 0 q13 q27 cghGain 0.339321 0.656687
CGHFrag 5 0 q14 q23 cghGain 0.473054 0.714571
CGHFrag 7 1 q11.1 q31 cghGain 0.362674 0.682635
CGHFrag 8 0 pter p12 cghLoss 0.000000 0.233533
CGHFrag 8 1 q11.1 q21.3 cghGain 0.343313 0.652695
CGHFrag 13 0 q11 q21 cghGain 0.189621 0.469062
CGHFrag 13 1 q21 q31 cghGain 0.600798 0.790419
CGHFrag 18 0 q12 q23 cghLoss 0.403194 0.976048
CGHFrag 19 0 pter q13.4 cghGain 0.000000 0.996008
CGHFrag X 0 pter q28 cghGain 0.000000 0.996008
CGHFrag Y 0 pter q11.2 cghGain 0.000000 0.582834
SkyCase "UCC_02 (internal Nr 08)" human 52 male <immunoType>
specimen
```

Figure 2.2: Habermann source data

2.1.1 The input Data

HyperTraPS requires data in the form of pairs of observations, a "before" and an "after" state. In this study, data used are the p and the q arm of the 23 pairs of chromosomes with CNAs. We used "+" to indicate if the arm of chromosome gain extra genetic traits and "-" for loss of some genetic traits. Such like 1p+, means the gain of genetic traits on the p arm of the first chromosome; and Xq- means the loss of genetic traits on the q arm of the X chromosome. This is a demonstration to show how the original data is transformed to the format applied to the model.

Sample 04 has a copy number gain on chromosome 3 running from q11 to q29 → 3q+
Sample 04 has a copy number gain on chromosome 9 running from pter to q34 → 9p+, 9q+
Sample 01 has a copy number loss on chromosome 8 running from pter to p12 → 8p-
Sample 01 has a copy number gain on chromosome X running from pter to q28 → Xp+, Xq+

In this way we can build up our binary set of traits for each sample. For those lines this would look like

Sample	3q+	8p-	9p+	9q+	Xp+	Xq+
04	1	0	1	1	0	0
01	0	1	0	0	1	1

In the dataset submitted from Habermann, there are 85 chromosomal arms assigning to CNAs. HyperTraPS model takes independent cross-sectional, longitudinal and phylogenetically related dataset and needs corresponding file format for applying to. The file format for independent cross-sectional dataset, which is used in this study, takes 0 at all positions of odd-numbered rows where 0 corresponds to that root. The root is assigned the initial states for all features of the independent cross-sectional dataset. Records on the aberrations are assigned to the corresponding positions on the even-numbered rows. Each column in the matrix presents one of the 96 possible positions indicating arms of the 23 pairs of human chromosomes. The number of row pairs corresponds to the number of samples applied to model. In this study, data matrix includes 124 row pairs corresponding to the 124 samples in Habermann's dataset. The following is an example of the file format.

```
0 0 0 0 0 0 0
1 0 0 1 0 1 0
0 0 0 0 0 0 0
1 0 0 0 1 0 1
0 0 0 0 0 0 0
0 1 1 0 0 1 0
```

2.1.2 TCGA dataset

The cancer genome atlas [Cancer Genome Atlas, 2012] is a international cancer genomic program resulting in the cooperation of researchers from diverse disciplines and multiple institutions of the world. TCGA study on colorectal cancer includes data on somatic copy-number alterations (SCNAs) from N=257 colorectal carcinoma DNA samples. For detecting the chromosomal aberrations, GISTIC algorithm was employed to identify significant peaks of amplification and deletion. In study, results from the focal and broad alterations of copy-numbers regions were published. The regions where recurrent focal alterations appeared are key regions where drive genes for cancer progresses are supposed located. Difference between both are that 21q- in focal but not in broad and 22q- in broad but not in focal. Result from the CGH employed observations is assumed more adequate to the broad aberrations. Hence the corresponding 22 features, which were identified in the broad aberrations in the TCGA study, in the dataset from Habermann's study were applied to running model. To apply a full dataset with 85 features from Habermann's study to HyperTraPS algorithm can be because of the huge number of parameters intractable.

2.1.3 Dataset for comparative study

I attempt to do a comparison study on the dynamic pathways of acquisitions resulting from H-CBN model and HyperTraPS for colorectal cancer. Hence the 11 features of chromosomal aberrations identified in [Gerstung et al., 2009] were applied to HyperTraPS algorithm to find the likely dynamic pathways.

2.2 HyperTraPS

HyperTraPS is a generalisable statistical platform to infer structure of dynamic pathways of biological traits [Greenbury et al., 2020]. The platform uses hypercubic transition pathways (HyperTraPS) to learn progression pathway from different types of observational data, longitudinal, phylogenetically related and independent cross-sectional. HyperTraPS was firstly introduced by [Johnston and Williams, 2016].

In this model, progressive dynamics are represented as paths on a hypercubic space connecting set of traits presence and absence. These patterns are represented by binary strings of length L, where 0 corresponds to the ith chromosome arm that are normal and 1 to those with CNAs in this study. A hypercubic transition network with edge weights W describing the probability of a transition between two states. It is assumed that all trajectories on the hypercubic transition network starting at the source state s_i makes for sure a transition to the target state t_i via any possible walks on the hypercube. Figure 2.3 is a copy of visual presentation of HyperTraPS model for cross-sectional data presents in [Greenbury et al., 2020].

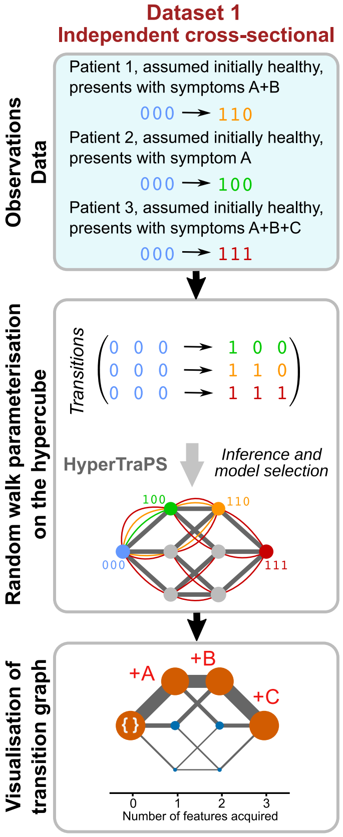

Dataset 1
Independent cross-sectional
Observations Data
Patient 1, assumed initially healthy, presents with symptoms A+B
000 → 110
Patient 2, assumed initially healthy, presents with symptom A
000 → 100
Patient 3, assumed initially healthy, presents with symptoms A+B+C
000 → 111
Random walk parameterisation on the hypercube
Transitions
0 0 0 → 1 0 0
0 0 0 → 1 1 0
0 0 0 → 1 1 1
HyperTraPS
Inference and model selection
100
110
000
111
Visualisation of transition graph
+A
+B
+C
{ }
0
1
2
3
Number of features acquired

Figure 2.3: Hypercubic cross-sectional data

2.2.1 HyperTraPS Algorithm

The HyperTraPS algorithm was first introduced by [Johnston and Williams, 2016] to sample random walk on a hypercube across a set of compatible states between a source and a target states. This ensure that no sampled paths involved in state cannot be reached. If starting from a given state, the sampled path is chosen proportional to its intensity at each step [Johnston and Williams, 2016] to ensures sampling preferential the most likely path. The set of transitions from the sources state s_i to the target state t_i showing in the observation dataset can be written as $D^{transition} = \{s_i \to t_i\}_{i=1}^{n_D}$.

In the algorithm, s_c is the source state and N_h is the number of sampled trajectories on a hypercube. α_i is the transition probability under parameterisation. The HyperTraPS algorithm can sample progression pathways efficiently that not all possible, but only the pathways crossing t-compatible states outgoing from sources state s are collected contributing

Algorithm 1: HyperTraPS algorithm for complete data

```
Data: $D^{transition} = \{s_i \to t_i\}_{i=1}^{n^D}$
Result: Estimate of $P(D^{transition} \mid W)$
begin
    for $(s \to t) \in D^{transition}$ do
        $s \leftarrow s_c$
        initialise $N_h$ trajectories starting at state $s$
        for $i \in N_h$ do
            $s_c \leftarrow s$
            $\alpha_i \leftarrow 1$
            while t-compatible move possible for $s_c$) do
                Calculate the probability of making a $t$-compatible move, record as $\alpha_i'$
                $\alpha_i \leftarrow \alpha_i \, \alpha_i'$
                Choose a $t$-compatible move at random in proportion to its transition
                  probability
                Make move and update $s_c$ accordingly
        $\hat{P}(s \to t) = N_h^{-1} \sum_i \alpha_i$
        $P(D^{transition} \mid W) \leftarrow P(D^{transition} \mid W) + \hat{P}(s \to t \mid W)$
```

to the likelihood calculation. As mentioned above, the likelihood of the probability density function based on the dataset D is the summation of all probabilities of observing a given transition, such that $L(W \mid D) = P(D \mid W)$. In the HyperTraPS algorithm, the value of likelihood of dataset under each parameterisation was collected to decide if a parameterisation is accepted. About 200 HyperTraPS trajectories,$N_h = 200$ was used to estimated likelihoods.

2.2.2 Bayesian framework

HyperTraPS has a Bayesian framework. Under this framework, parameters for the set of edge weights W on the hypercubic cubic are inferred from observation data D.

$$P(W \mid D) = \frac{P(D \mid W)}{\int P(D \mid W)\, P(W)\, dW} P(W)$$

The main quantity of interest is the posterior probability $P(W \mid D)$ referring to probability of transition steps between states on the hypercubic. The prior $P(W)$ is the distribution of weight that we imposed on parameters of model. In this study, we imposed no assumption on the model parameters, so we set uniform distribution to the prior $P(W)$ in this study. $P(D \mid W)$ in the formula is the likelihood of the probability density function based on the dataset D, such that $L(W \mid D) = P(D \mid W)$. The integral of the likelihood and the prior, $\int P(D \mid W)\, P(W)\, \mathrm{d}W$, is then a fixed value. In Bayesian framework, the posterior probability is then only proportion to the product of likelihood and the prior. That means, to drive samples from the posterior distribution, we need only set criteria on the likelihood calculation.

2.2.3 Hidden Markov Chain (HMM) modelling

HyperTraPS has a Markov chain modeling of the edge weight of transitions between different states. This is in fact a hidden Markov chain (HMC). Some hidden process, such like signals randomly emitted by the walkers and a signal corresponds to the current set of acquired traits of the random walker, must have happened before to arise the observation [Greenbury et al., 2020]. Under this assumption, the probability of observing of a given transition requires a signal transmitted by both source and target states and signals signify the system have reached source state and then made the transition to the target state via any possible random walks on the hypercube.

MCMC sampling

There are two properties included in MCMC: Markov chain and Monte-Carlo method. In this study, we wanted infer the temporal orders of genetic traits CNAs acquired in the cancer progression, the main quantity of interest is the transition probability between states of acquisition. This is the posterior distribution in the Bayesian framework. As we mentioned above, the posterior $P(W \mid D) = \frac{P(D|W)}{\int P(D|W)\, P(W)\, dW} P(W)$. We can perhaps direct calculate $P(D \mid W)P(W)$, but the calculation of integral $\int P(D \mid W)$ is in a high-dimensional space, as in this study, difficult. Therefor we need a sampling method like MCMC to sample the posterior in case that all we know is how to calculate the likelihood.

An alternated Metropolis-Hastings algorithm is built in the MC sampler aiming to ensure that the stationary distribution we choose approximates the target distribution in the study.

Alternate Metropolis-Hastings algorithm is a modification of the basic Metropolis algorithm. For our desired or target distribution in this study, the posterior $P(W \mid D)$, the Metropolis-Hastings algorithm can draw samples from its distribution if we can provide the calculation of the likelihood $P(D \mid W)$, which is proportion to the density of the posterior probability distribution.

Supposed for our parameter θ, our target distribution is $P(\theta)$. We initial a starting distribution $p(\theta^0)$ for the starting point θ^0. We propose a jumping distribution, $J_t(\theta^* \mid \theta^{t-1})$ that suggests a candidate for the next sample θ^* at time t given the previous sample value θ^{t-1} at time $t-1$. The proposal distribution is symmetry, namely $J_t(\theta^* \mid \theta^{t-1}) = J_t(\theta^{t-1} \mid \theta^*)$. For each iteration, we generate a candidate θ^* from the distribution $J_t(\theta^* \mid \theta^{t-1})$ (the Markov property of the draws). We calculate the ratio:

$$r = \frac{\frac{p(\theta^*)}{J_t(\theta^*|\theta^{t-1})}}{\frac{p(\theta^{t-1})}{J_t(\theta^{t-1}|\theta^*)}}$$

which is used to decide if we accept the candidate or not.

$$\theta^t = \begin{cases} \theta^* & \text{with probability min } (r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

r here is only the ratio of the probability of the sequential draws, $\frac{p(\theta^*)}{p(\theta^{t-1})}$ form our target distribution because $J_t(\theta^* \mid \theta^{t-1}) = J_t(\theta^{t-1} \mid \theta^*)$ [Gelman, 2013]. If the iterations runs long enough, r closes to 1 and this indicates that ratio of pair of sequential draws are identical. This means the Markov chain consisting of collection of samplings comes to stationary that all draws from the chain has the same stationary probability.

Monte-Carlo approximation

The calculation of the mean of posterior $P(W \mid D)$ probability in this study involves in an integral in multi-dimensional space because parameter $W = \{w_1, w_2, ...w_n\}$, weight of edge for n edges is multi-dimensional. In the Bayesian framework, to get the mean of W , we need integral in multi-dimensional space, $\bar{W} = \int WP(W \mid D)dW$. This calculation seems intractable and cannot be solved analytically. Monte-Carlo method provides a way to calculate its mean and variance [Wasserman, 2004].

The integral is the mean of the posterior, can be, like any other integral, to be unwritten to a integral of a uniform distribution over range (a, b) and others.

$$I = \int_a^b h(x)dx = \int_a^b w(x)f(x)dx$$

where $w(x) = h(x)(b - a)$ and $f(x) = \frac{1}{b-a}$. f is the probability density for a uniform distribution over (a, b). By the law of large numbers, the estimand of the integral is then the expected value of the posterior. It is also the mean value of all collected samples.

$$I = \mathbb{E}_f(w(X))$$

where X is a uniform distribution over range (a, b). By the law of large numbers:

$$\hat{I} \equiv \frac{1}{N}\sum_{i=1}^{N} w(X_i) \to \mathbb{E}_f(w(X) = I.$$

The standard derivation of posterior is:

$$\hat{se} = \frac{s}{\sqrt{N}}$$

where

$$s^2 = \frac{\sum_{i=1}^{N}(w(X_i) - \hat{I})^2}{N - 1}$$

APM MCMC embedding of HyperTraPS A sampler auxiliary pseudo-marginal Markov Chain Monte Carlo (APM MCMC) [Andrieu and Roberts, 2009, Murray and Graham, 2016] is embedded in the HyperTraPS algorithm to produce parameter of interest, the posterior distribution, in the study. Applying this algorithm can improve the mixing of sampler that results in poor mixing when hamming distance between the source and the target state become large. This algorithm introduce a new variable u to model and make the likelihoods as a joint density $l(\pi, u)$. π is the maximal likelihood parameterisation of model. Updating the Markov chain is then performed by keeping π and u alternately fixed. In HyperTraPS we draw our estimate of likelihood from the set of random trajectories, the proposals for the new variable u, across hypercubic. The APM MCMC satisfies the same convergence property as MCMC and this embedding of APM MCMC in algorithm enable HyperTraPS likelihood estimation for long pathway calculation [Greenbury et al., 2020].

Model selection

The log-likelihood collected in the algorithm will be used to calculate Akaike Information Criterion (AIC) score to identify the sparsest model that has the largest AIC value. $AIC = 2(k - \hat{l})$, where k are the number of parameters in the model, and $\hat{l}$ is the maximal log-likelihood. Model of cross-sectional independent sample, as we have in this study, the regularised model needs $k = L$, L is the number of feature. Hence, the best model has the largest AIC score, as well as the largest log-likelihood value.

Gelman-Rubin Test

To evaluate the convergence of MCMC sampling is an important to validate our draws. To do inference of draws from iterative simulation, there are two possible problems that we can encounter. One problem is the possibility that we did not get representative to target function because of running the iterative process not long enough. This could let our further processing on statistical inference or model building to be in question.

The other problem is the existence of within-sequence correlations. Draws from correlated samplings are less precise then those from independent samplings. Consequence of this situation is a large amount of draws do not lead to proper approximating the target distribution. Gelman-Rubin test [Gelman and Rubin, 1992, Brooks and Gelman, 1998] is applied to test the convergence of the draws to ensure the well-mixing of sequences of draws and that sequences reached stationarity.

To apply sequences of draws to the Gelman-Rubin test, there are some requests on them [Gelman, 2013]. we should take some sequences in which warm-up period have been discarded. This is already done in the HyperTraPS algorithm. Only the log-likelihood of parameterisation after burn-in were printed out after running model. To ensure the well-mixing, we need to apply at least 2 sequences that start from various places to the test. We calculate at first the within- and the between sequences variance [Gelman, 2013]. For each estimand ψ, we label the simulations as $\psi_{ij} (i = 1, \ldots, n; j = 1, \ldots, m)$. i is the number of draws in each sequence and j is the number of sequences. We can computer the between sequences variances B and the within-sequences variances W:

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\overline{\psi}_{.j} - \overline{\psi}_{..})^2, \text{ where } \overline{\psi}_{.j} = \frac{1}{n}\sum_{i=1}^{n}\overline{\psi}_{ij},\ \overline{\psi}_{..} = \frac{1}{m}\sum_{j=1}^{m}\overline{\psi}_{.j}$$

$$W = \frac{1}{m}\sum_{j=1}^{m}s_j^2, \text{ where } s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\psi_{ij} - \overline{\psi}_{.j})^2$$

We can estimate the variance of the posterior probability, $var(\psi \mid y)$ by a weighted average of W and B:

$$v\hat{a}r^{+}(\psi \mid y) = \frac{n-1}{n}W + \frac{1}{n}B$$

In general, this quantity overestimates the marginal posterior variance because we assume the starting distribution is appropriately overdispersed. This quantity should be unbiased under stationarity, or if $n \to \infty$.

In the limit as $n \to \infty$, the between sequences variances B will no more exist, and the $var(\psi \mid y)$ approach variance of W. That means for any finite n, the variance of W is an underestimate of $var(\psi \mid y)$. An important indicator of convergence of the simulation is the estimated:

$$\hat{R} = \sqrt{\frac{v\hat{a}r^{+}(\psi \mid y)}{W}}$$

This value, the potential scale reduction factor (PSRF or R_c), is bigger than 1 for any finite n and will approach 1 if $n \to \infty$. In general, the simulation is supposed to be convergent if $R_c < 1.2$ for all parameters, or more string condition $R_c < 1.1$ [Gelman and Rubin, 1992, Brooks and Gelman, 1998].

Some aspects on HyperTraPS algorithm

There are many different kinds of jumping distributions designing for efficient simulation. Some features of algorithm are important to enable efficient simulation, such like that each jump goes a reasonable distance in the parameter space, otherwise the random moves too slow; the jumps should not be rejected too frequently otherwise random walk waste too much time to stand still. Since speed of simulation important for efficiency, to "thin out" sequence, namely instead taking every draw simulated, only every some numbers of draws to speed up simulation process. In the HyperTraPS algorithm, the proper random seed numbers were found out through repeated investigations applying $N_c = 10$ to algorithm ; the in the warm-up period, about 20% of total, were discarded ; only draws from every 10^3 iterations are taken contributing to calculation [Johnston and Williams, 2016]. The burn-in period of HyperTraPS is at around $2x10^5$ iterations.

2.2.4 Inference

The output file includes a simple summary of the dynamics inferred by HyperTraPS: the probability that each feature is acquired/lost at each possible ordering. Different presentations are applied to infer the potential transitional pathways across the hypercube.

Simulation of Random Walk

To find the potential order of feature acquisition in the colorectal cancer progress, we simulated only trajectories corresponding to transitions that are observed in the dataset. This graph presented the probabilities of all transition steps from the initial states, denoted 0 in this study, to the target state t. We can record the feature i acquired at step j as f_{ij}. In the cancer progresses, all observed features are always gained at all stages of the simulation process, therefore the property $\sum_{k=1}^{j} f_{ik} = 1$ and $\sum_{k=1}^{i} f_{kj} = 1$ holds To perform the feature i acquired at step j, we can consider f_{ij} as the probability:

$$f_{ij} \approx P(\text{ feature } i \text{ is gained at step } j \mid s = \{0\}^L \rightarrow t = \{1\}^L)$$

where s is the source state and t the target state of the set of random walks [Greenbury et al., 2020].

Constraint of transition pathway in cancer progresses

The acquisition of features is a irreversible process that each feature was only acquired on one of the step in the whole process. For example, in a $L = 3$ HyperTraPS model, the steps from state (000) to state (110) could be consisted of transition $000 \rightarrow 100$ and transition $100 \rightarrow 110$ or of $000 \rightarrow 010$ and $010 \rightarrow 110$. In the model, each transition has a given *intensity*, and the probability of a given transition from some state is proportional to that transitions intensity [Johnston and Williams, 2016]. If we expected 90% of process to follow the initial step 000 → 010. Then it is highly likely to be 000 → 010 than to be 000 → 100. The most likely transition pathway can be calculated by applying sequentially the largest f_{ij} from step 1 to the last step. R script is written to apply to do the evaluation.

Probabilistic feature graph representation

One effective method chosen to do inference is to visualize the transition to a directed weighted acyclic graph (weighted DAG) through state space, where the acquisition of features is a irreversible process. Graphical models are usually used to describe the dependence/ independence relationships between random variables in the Bayesian framework. A graphical model contains nodes and (directed or in-directed) edges. Each node in the graph corresponds to a feature with the structure of edge of the graph, such like the weighted edge. The structure of edge determines the conditional dependence relationship between features. For any DAG constructed by random variables x with k nodes, we have

$$p(x) = \prod_{k=1}^{K} p(x_k \mid pa(x_k))$$

where $pa(x)$ denotes the "parents" of node x_k. This implies, for all k we have $p(x_k \mid x_1, \ldots, x_{k-1}) = p(x_k \mid pa(x_k))$. This means the joint conditional distribution can be simulated when some nodes are observed.

The graph includes potential, also the most likely, pathways of acquisition in the cancer progresses. The dynamic of the acquisition was also presented in a summary graphs of the amount of features acquisition at each ordering and the corresponding table.

Comparison with other studies

Directed graphical model are known as Bayesian networks. I took a comparison with studies [Gerstung et al., 2009, Gerstung et al., 2011] that used hidden conjunctive Bayesian network (H-CBN) presenting the dependence relationship between different genetic traits in colorectal cancer. An important different between HyperTraPS model and H-CBN model is the interpretation of the joint conditional distribution between features. H-CBN imposed a monotonic relationship between features and their ”parents” features. The HyperTraPS model relaxes the impose and allows the multiple orderings of features. Therefore the joint conditional distribution between features are interpreted as the magnitude of mutual influence between features. The difference of those two models leads also to the number of potential pathways can be derived from models [Greenbury et al., 2020].

Chapter 3

Results and Analysis

The [output file].process contains a simple summary of the dynamics inferred by HyperTraPS came out after running an instance. File contains 5 columns with the following messages: [ordering index] [feature index when sorted by mean ordering] [original feature index] [feature label] [probability]. Column 5 contains the probability that each feature is acquired/lost at each possible ordering. This is the probability of transition of feature s, $P(Y_{out}, X_{in}; s)$, that feature Y is acquired leaving state s, with feature X having been acquired to reach state/feature s.

3.1 Features transitional dynamics

A SVG graph plots original feature index in column 3 horizontally, ordering index in column 1 vertically, and probability in column 5 as colour or point size of the [output file].process presenting the amount of features acquired at each ordering. Graph beneath is the output plot after employing the 22 features which were identified as colorectal associated in the TCGA study, in Habermann's dataset to run HyperTraPS.

X-axis labels indices of the 22 features applied to study and their corresponding aberrations of chromosomal arms. Y-axis labels the order of feature/state from which transition path went out. For independent cross-sectional dataset, the first state all random walks went out is the root labelled order 0. Applying to a dataset with 22 features to simulate a random walk, random walk went across 23 feature/state and 22 orderings at the hypercubic. Using the analysis code (see code B.11) on the output file, we can calculate the transition probability for each feature at each ordering. The score of the transition probability in table 5.1 , table 5.2 corresponds to the radius of circle assigned to each ordering on the output plot.

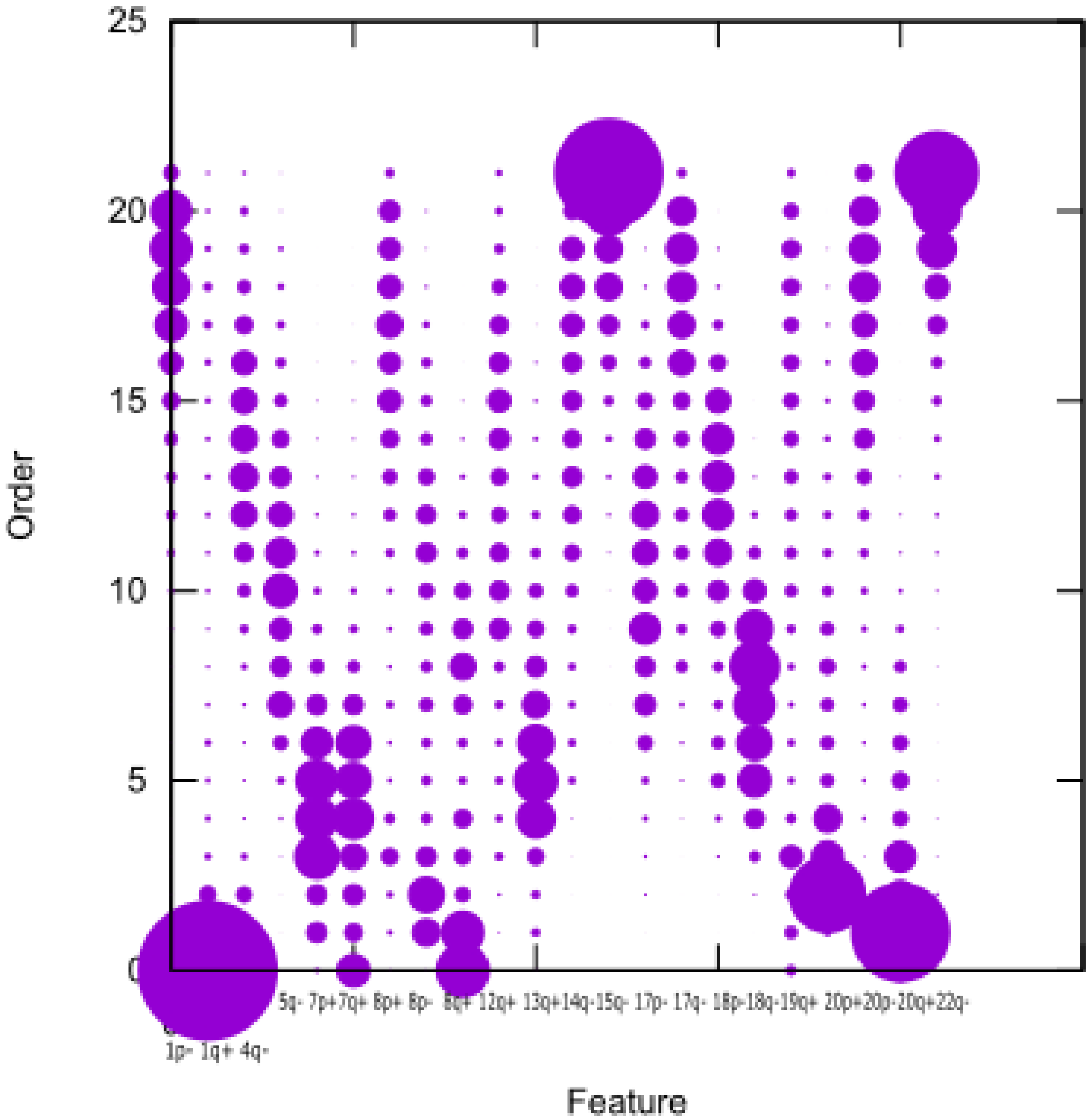


Figure 3.1: Heatmap style graph of acquisition probability

The corresponding feature ordering posterior which shows how much features is acquired at each ordering.

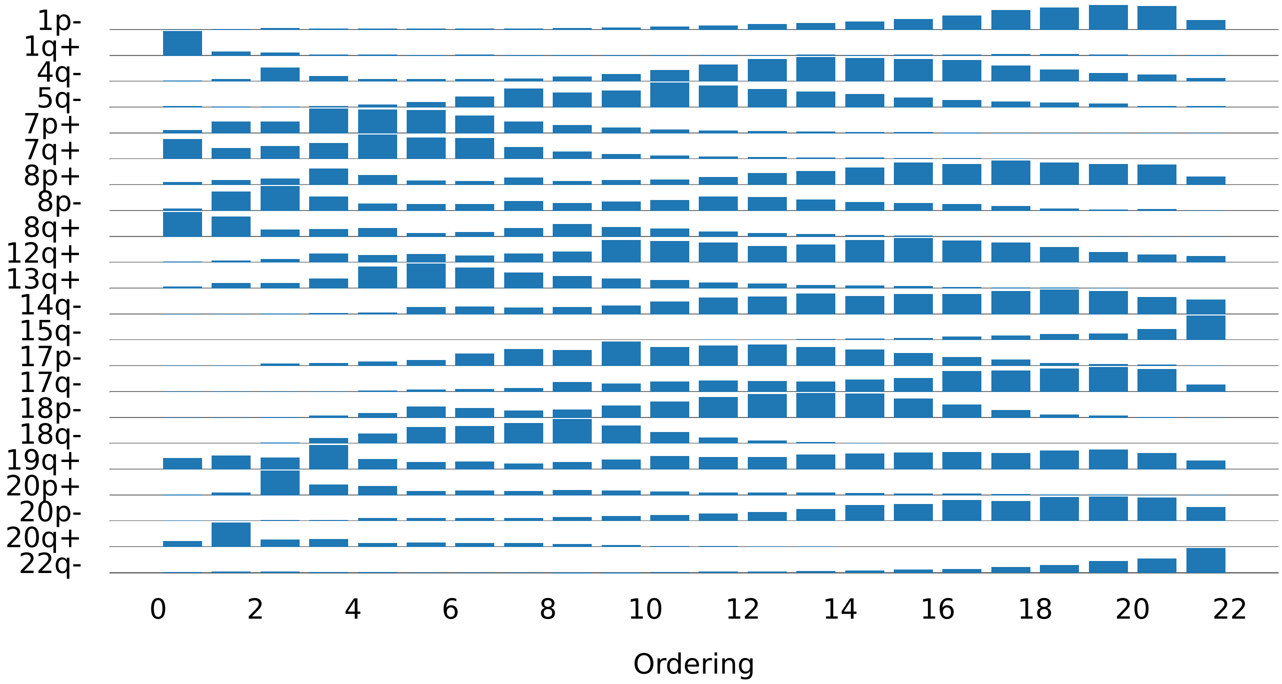


Figure 3.2: Application to 22 features

3.1.1 Inference: the most likely temporal order

We are interested in applying the posterior output to find the most likely temporal order of feature acquisition in the cancer progression. Order begins at root. Using the analysis code on the (see code B.12), we chosen the feature with the highest transition probability at ordering 0 , then sequentially to the last ordering 21. If a feature is chosen, we set value -1 to all column of this feature. It is assumed that cancer progression is a process of feature accumulation. If feature once acquired, will not compete with feature acquisition in the progression. Because the probability of competing on acquisition is bigger than 0, therefor we set -1 to orderings of features that were already acquired to disqualify them in competition. Figure 3.3 presents the most likely temporal order of feature with thick line. Fine lines at the background is the second most likely temporal order of feature. We choose the second highest transition probability at each ordering and add the one not chosen to the end of order. Graph is created by way of applying matrix including the first three most likely temporal order to igraph in R. Edge width correspond to the weight, transition probability, of edges. For be able to present clearly enough the most likely order, I applied weight 1 to edges associated to the most likely order.

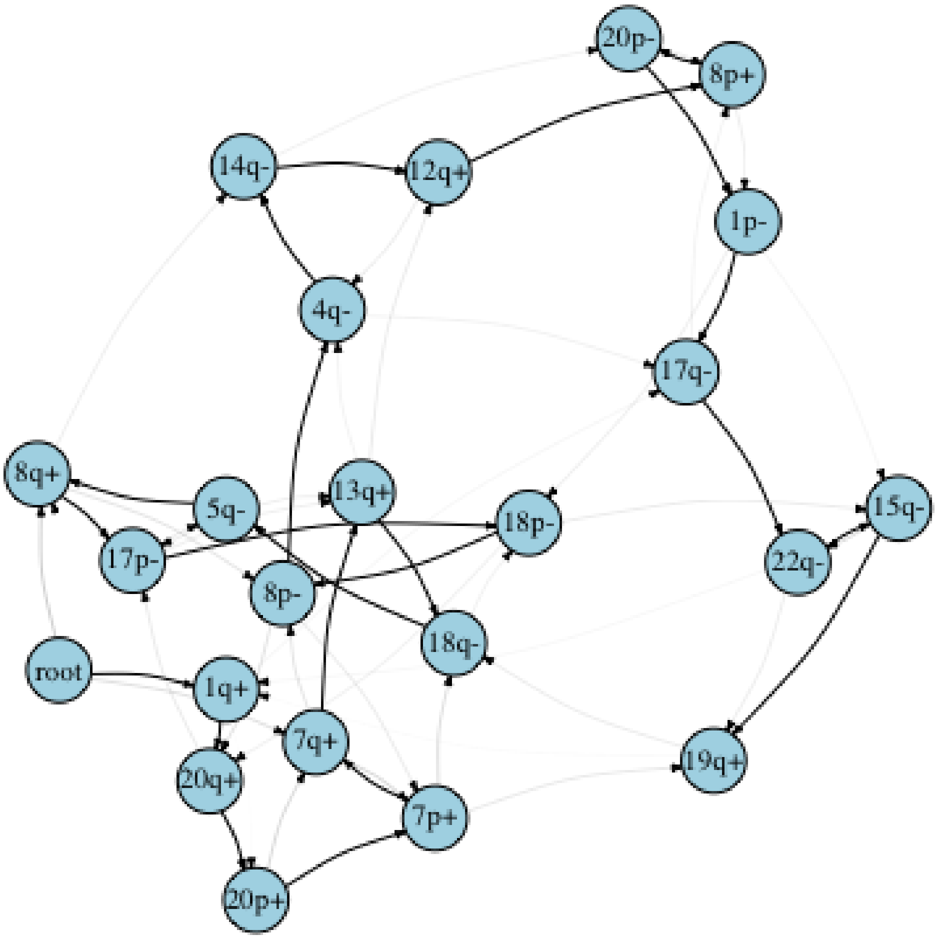


Figure 3.3: Likely pathways, 22 features identified in TCGA study

3.1.2 mean and SD of mean of features acquisition

As the random walk went across the hypercubic space, features were acquired at each ordering. Numbers of features acquired at each ordering is the product of the times state encountered and the probability of transition of state from which feature went out. The expected total number of features acquired is then the summation of features acquired at all orderings in the random walk.

$$P(Y_{out}, X_{in}; s) = \sum_{s} P(Y_{out}, X_{in}; s)P(s)$$

where feature Y is acquired leaving state s, with feature X been acquired to reach state s. $P(s)$ is the proportion the state s is encountered.

For independent cross-sectional dataset, numbers of features acquired is 0 because times state encountered is 0. The times state encountered for the state/feature 1 is then 1, for state 2 is 2 etc. The mean number of features acquired presents how far a feature can be acquired in the state space. Feature with a small mean number means that feature was acquired at short distance away from root and big number at far distance away from root. The standard derivation (SD) of mean of features acquisition presents how dispersed features acquired order. If the SD of mean is small, features acquired order centred, otherwise is more dispersed. Using the analysis code on the output file, [output file].process, the scores of mean and SD of features acquisition of the 22 genetic traits in Habermann's dataset were calculated and sorted from high to low presenting in table 3.1.

Table 3.1: Mean number of feature acquired, 22 features

	Chromosome	Mean.of.FA	SD.of.FA
1	15q-	0.8509	1.7992
2	22q-	0.7977	1.4921
3	1p-	0.7389	0.9671
4	20p-	0.696	0.7551
5	17q-	0.6826	0.7269
6	14q-	0.6508	0.5791
7	8p+	0.6018	0.5948
8	4q-	0.5723	0.5346
9	12q+	0.5575	0.4277
10	18p-	0.5231	0.539
11	19q+	0.51	0.4015
12	5q-	0.4911	0.4108
13	17p-	0.4856	0.4256
14	8p-	0.3657	0.2596
15	18q-	0.3339	0.4446
16	13q+	0.3056	0.2662
17	20p+	0.2798	0.1517
18	7p+	0.2339	0.2378
19	8q+	0.228	0.23
20	7q+	0.2202	0.2168
21	1q+	0.2189	0.153
22	20q+	0.1557	0.1388

3.1.3 Result Analysis

From the figure 3.2, we can see the dynamic structure of the acquisition of the 22 CRCs associated features identified in TCGA study. The cancer progression seems a continuous acquisition of features. Some, like 1q+, 20q+, 20p+ were obvious mostly acquired only in short time at the beginning and 15q-, 22q- at the end of the cancer progression. In addition, some other amplified features, like 7p+,7q+,8q+.13q+ were acquired mainly on the first part of progression. Features mainly acquired at the last one third part of progression were those with deletion, 1p-,17q-.

That there exist a pathway with amplified features, ”+” and another consisting of features with ”deletion”, ”-”, and overlapping of those two pathways were claimed in some studies of CRCs [Höglund et al., 2002, Hjelm et al., 2006]. The pathway dynamic graph cannot give a direct confirmation on such claim. Likewise, graph shows the decreasing of acquisition of some amplified features followed by the slow acquisition of big amount of features of both amplified and deleted. To the end comes out the abrupt increased acquisition of only deleted features.

3.2 Model Comparison

In the study, [Gerstung et al., 2009], Gerstung apply a Hidden-Conjuctive Bayesian Network (H-CBN) to infer the temporal order of chromosomal aberrations in the progression of cholorectal cancer. He applied 11 traits of chromosomal aberrations associated to CRCs, comparative fewer than 22 identified in the TCGA study. In my study, I applied the same 11 traits to run HyperTraPS model. The first three graphs beneath were derived from HyperTraPS. The last graph was presented in [Gerstung et al., 2009].

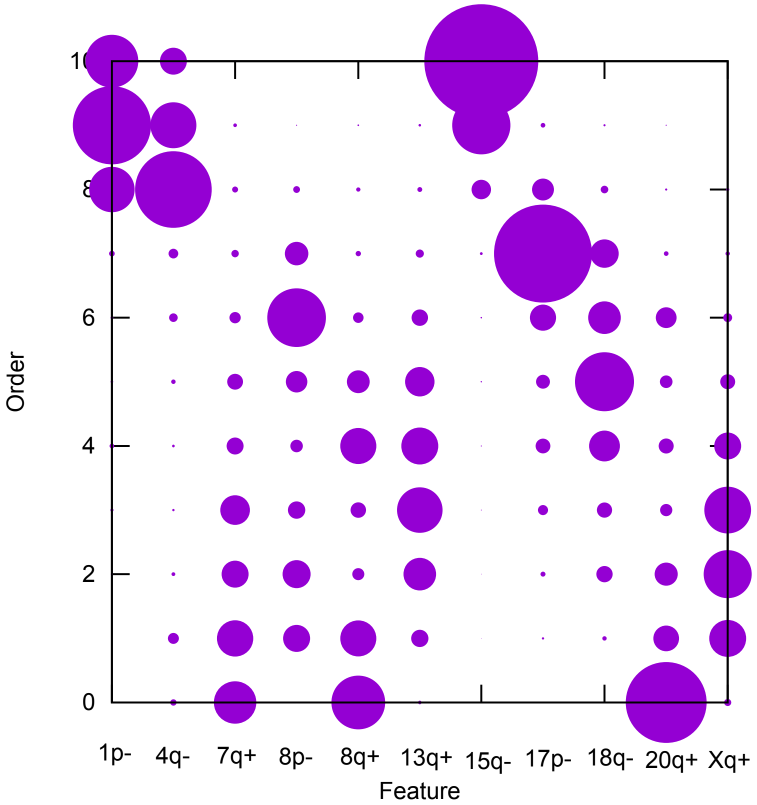


Figure 3.4: Heatmap-style output of transition probability

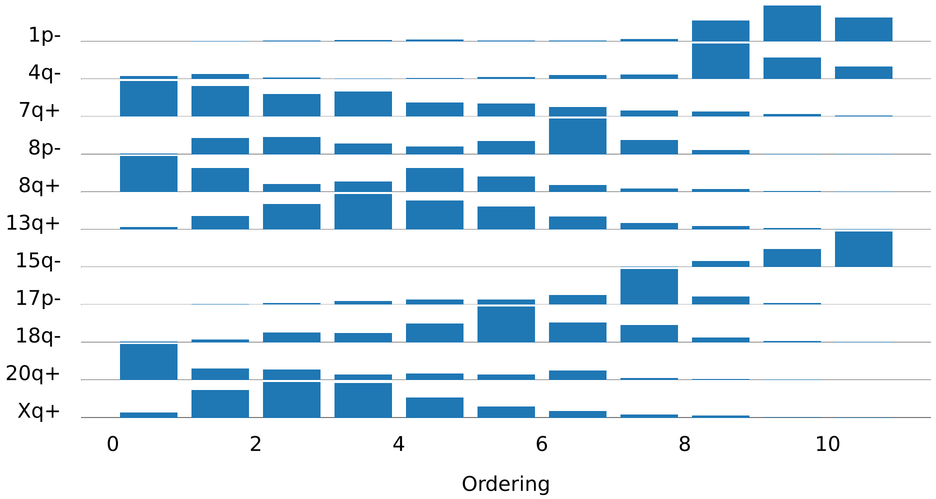


Figure 3.5: The HyperTraPS inferred structure of feature acquisitions of CRCs, 11 features

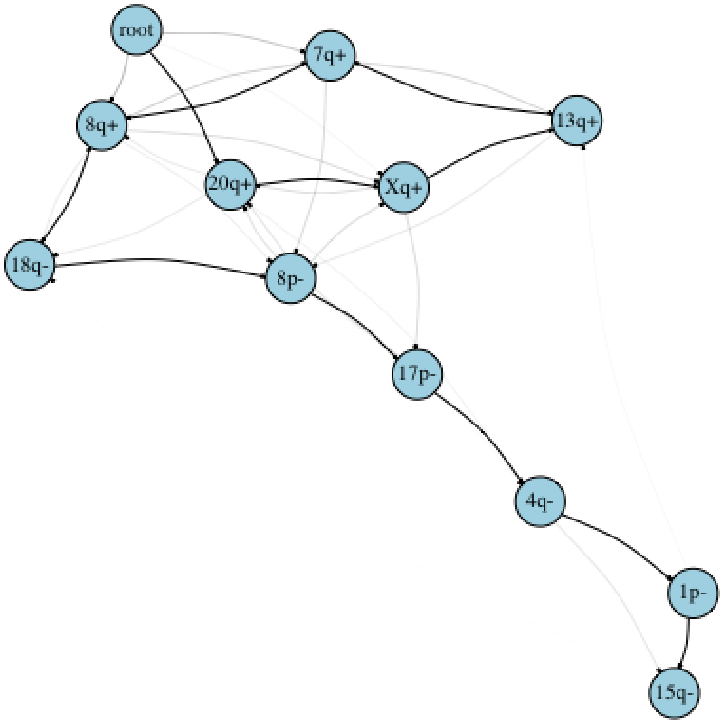


Figure 3.6: The most likely colorectal cancer progresses

Figure 3.5 shows amount of feature acquisition at each ordering. The likely temporal order of the 11 identified features in [Gerstung et al., 2009] learned from HyperTraPS presents in graph 3.6. As mentioned above, edge width correspond to the weight, transition probability, of edges. Weight of edges associated to the most likely order applied to is 1. Figure 3.7 is the result of applying to H-CBN model to the 11 identified features in [Gerstung et al., 2009].

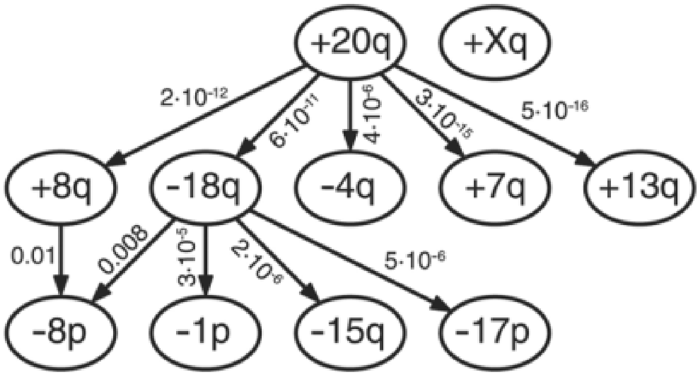


Figure 3.7: H-CBN inferred structure of CRCs feature acquisitions

The following table includes a comparison on probabilistic graphs of CRCs cancer progression derived from HyperTraPS and H-CBN model.

Table 3.2: Comparison HyperTraPS vs H-CBN

Model	HyperTraPS	H-CBN
DAG nr.	many	one
edge	no imposed causal	imposed causal
feature Nr.	up to 65	under 20
first feature	7q+,8q+,20q+	20q+,13q+
feature acq.pattern	similar: start ”+”; end ”-”	

Chapter 4

Discussion and Conclusion

The HyperTraPS models the feature acquisition as a Markov chain and performs better than study [Hjelm et al., 2006]. With Bayesian framework and MCMC sampling, we only have to consider the likelihood function of parameterisation and the process of parameter estimation is faster than the maximal likelihood estimation. The parameter estimation using MCMC sampling is faster because not the exact but the approximate likelihood is calculated for updating chain. Therefore can HyperTraPS tackle much more large L in model than the number of module in [**?**]. The computation time for a modeling with simple Markov chain usually scales exponentially with the number of traits.

Comparing to study that models dynamic pathway as a hidden Markov chain (HMM) , such like the H-CBN from [Gerstung et al., 2009, Gerstung et al., 2011]. In HyperTraPS, an observation of chromosomal aberrations is supposed the underlying transition has reached the corresponding state, therefore there is no observation errors calculation in model. In model, only the trajectories that has been reached the target states will be sampled. It is not necessary to do extra calculation for such parameter before the selection of best model basing on likelihood values. On the contrary, the H-CBN model includes parameters for the error calculation. Therefore, in calculation of the best model, an EM algorithm must first introduced to get the estimand for error parameters before applying another algorithm to find the best parameterisation of model. But still, HyperTraPS shows the ability to infer similar temporal order of feature acquisition as employing the model H-CBN.

The Bayesian framework enables HyperTraPS choose different prior according to the need modelling different situations. Prior is used to impose constraint on the relationship between parameters in the model. HyperTraPS model has proof to be able to include tree like precedence prior to influence the dynamic process of dataset in the hypercubic space [Greenbury et al., 2020]. This make the model more flexible for different kind of dataset, those we have belief their structure and those we have no. For this study, a uniform prior was applied that not constraint on the relation of parameter was imposed. It could be interesting to apply precedent prior on base of established knowledge from clinical studies on CRCs to the model. This a big advantage of HyperTraPS comparing to other existing models on CRCs progression.

Similar to model in [Hjelm et al., 2006], it is assumed that all traits of the first order HyperTraPS model, which is suitable for cross-sectional independent samples, have the same dependence. In this study, there is total 492 observations applied to HyperTraPS model with 22 features. It is assumed that we do not run the risk of over-fitting because the parameter number needed is only 22 parameters.

In general, results we get from this study can give a good explanation to various findings in existing studies on CRCs progression. It can give some explanation to the finding in [Höglund et al., 2002] on the 2 distinct temporal orders of group of "gain" and of "loss" features and the overlapping of those two in CRCs progression. About the positive and negative correlated between groups of features in [Hermsen et al., 2002] can be explored in the future applying elaborated design to HyperTraPS model. The finding in the study can provide some information about features grouping to refine design for study on CRCs progression in the future.

Chapter 5

Table

Table 5.1: Transition probability of features at different orders from the cancer genome atlas (TCGA) dataset 1.

	Chr	order0	order1	order2	order3	order4	order5	order6	order7	order8	order9	order10
1	1p-	7e-04	0.0041	0.0093	0.0073	0.0063	0.0065	0.0086	0.0065	0.0098	0.0139	0.0191
2	1q+	0.4797	0.0826	0.0638	0.024	0.0206	0.0178	0.0231	0.0155	0.0121	0.0112	0.016
3	4q-	0.0052	0.0104	0.0584	0.0239	0.011	0.0105	0.0112	0.0136	0.0219	0.0329	0.0477
4	5q-	0.0074	0.0042	0.0046	0.0063	0.0133	0.0273	0.0541	0.0932	0.0744	0.0823	0.1215
5	7p+	0.0196	0.076	0.0742	0.1583	0.1513	0.1504	0.1129	0.0749	0.0533	0.0361	0.0257
6	7q+	0.1186	0.0667	0.0774	0.0957	0.1452	0.1276	0.1242	0.0725	0.045	0.0302	0.0225
7	8p+	0.0115	0.0197	0.0239	0.062	0.0375	0.0168	0.0153	0.0277	0.0146	0.0181	0.0201
8	8p-	0.011	0.1005	0.1284	0.0735	0.0383	0.0356	0.035	0.0499	0.0417	0.0481	0.0566
9	8q+	0.1874	0.1523	0.0554	0.0602	0.0678	0.0295	0.035	0.0677	0.0955	0.0733	0.0611
10	12q+	0.0041	0.0074	0.0128	0.031	0.0266	0.0296	0.0256	0.0317	0.0381	0.0776	0.0742
11	13q+	0.0118	0.0331	0.0325	0.0623	0.1388	0.155	0.1319	0.0985	0.0765	0.0613	0.0526
12	14q-	9e-04	7e-04	0.0026	0.0044	0.0062	0.0271	0.0288	0.0252	0.0273	0.0322	0.0463
13	15q-	6e-04	3e-04	6e-04	9e-04	0.0011	0.0014	0.0017	0.0022	0.0036	0.0034	0.0051
14	17p-	0.0022	0.003	0.0111	0.0131	0.0202	0.0276	0.0567	0.0779	0.0736	0.1122	0.0863
15	17q-	5e-04	8e-04	0.0013	0.0021	0.0063	0.0104	0.0125	0.0181	0.0449	0.0369	0.0476
16	18p-	8e-04	0.0017	0.0034	0.0115	0.0216	0.0532	0.0465	0.0341	0.038	0.0571	0.0769
17	18q-	2e-04	0.0013	0.01	0.0391	0.0718	0.1187	0.1259	0.1483	0.1776	0.1328	0.0834
18	19q+	0.0391	0.0475	0.0397	0.0836	0.0361	0.0244	0.027	0.0205	0.0256	0.0331	0.0452
19	20p+	0.0069	0.0294	0.2664	0.1144	0.1004	0.0483	0.0499	0.0476	0.0581	0.0496	0.0387
20	20p-	0.0029	0.0024	0.0048	0.0066	0.0156	0.0153	0.0139	0.0156	0.0194	0.0228	0.027
21	20q+	0.0822	0.3427	0.1053	0.1126	0.057	0.0618	0.0564	0.0536	0.0433	0.0288	0.0168
22	22q-	0.0065	0.0132	0.014	0.0069	0.0069	0.0051	0.0037	0.0051	0.0059	0.0059	0.0096

Table 5.2: Transition probability of features at different orders from the cancer genome atlas (TCGA) dataset 2.

	Chr	order11	order12	order13	order14	order15	order16	order17	order18	order19	order20	order21
1	1p-	0.0272	0.0358	0.0401	0.0504	0.0653	0.0854	0.1183	0.1335	0.1491	0.1449	0.0578
2	1q+	0.0154	0.0192	0.021	0.0191	0.0218	0.0233	0.0298	0.0316	0.0225	0.0167	0.0134
3	4q-	0.0715	0.0952	0.1031	0.0982	0.0955	0.09	0.0671	0.0506	0.0367	0.0307	0.0145
4	5q-	0.1084	0.0911	0.0775	0.0655	0.0478	0.0368	0.0284	0.0241	0.0191	0.0069	0.0058
5	7p+	0.019	0.0134	0.0103	0.008	0.0068	0.0044	0.0026	0.0019	4e-04	1e-04	1e-04
6	7q+	0.0167	0.0135	0.0107	0.0097	0.0082	0.0066	0.0042	0.0024	0.0017	4e-04	1e-04
7	8p+	0.0312	0.0449	0.054	0.0671	0.0853	0.0806	0.0935	0.0856	0.0806	0.0777	0.0323
8	8p-	0.0749	0.0709	0.0597	0.0449	0.041	0.0362	0.0252	0.0121	0.0067	0.0091	8e-04
9	8q+	0.0397	0.0281	0.021	0.0136	0.0077	0.0027	0.0017	5e-04	1e-04	0	0
10	12q+	0.0695	0.0579	0.0627	0.079	0.0853	0.0764	0.0691	0.0538	0.0365	0.0275	0.0236
11	13q+	0.0371	0.0297	0.0218	0.0191	0.0144	0.0099	0.0055	0.0042	0.003	6e-04	3e-04
12	14q-	0.0612	0.0653	0.0757	0.0676	0.0743	0.075	0.085	0.0904	0.0861	0.0634	0.0543
13	15q-	0.0083	0.0128	0.0187	0.0249	0.037	0.0564	0.0738	0.0986	0.1005	0.1718	0.3762
14	17p-	0.0926	0.0988	0.0862	0.0758	0.06	0.0422	0.0291	0.0132	0.0086	0.0069	0.0029
15	17q-	0.0518	0.0497	0.0465	0.0555	0.064	0.0945	0.0971	0.1082	0.1135	0.1041	0.0335
16	18p-	0.0979	0.1106	0.1151	0.1128	0.0903	0.0612	0.0366	0.0153	0.0106	0.0046	2e-04
17	18q-	0.0453	0.0241	0.0115	0.0056	0.0026	0.0011	5e-04	1e-04	0	0	0
18	19q+	0.0419	0.0428	0.0503	0.0544	0.0566	0.0593	0.0557	0.0639	0.0678	0.0556	0.03
19	20p+	0.0319	0.0299	0.0297	0.0244	0.0207	0.0186	0.0144	0.0101	0.0074	0.002	0.0011
20	20p-	0.0338	0.0404	0.0547	0.0737	0.0766	0.0948	0.0912	0.1088	0.1103	0.1063	0.0629
21	20q+	0.0119	0.0095	0.0065	0.0035	0.0029	0.0021	0.0013	8e-04	3e-04	6e-04	0
22	22q-	0.0129	0.0165	0.023	0.0272	0.0359	0.0426	0.0698	0.0905	0.1384	0.17	0.2902

Table 5.3: Transition probability of features at different orders from the H-CBN dataset.

	Chromosome	ordering0	ordering1	ordering2	ordering3	ordering4	ordering5	ordering6	ordering7	ordering8	ordering
1	1p-	0.0017	0.0051	0.0092	0.0152	0.0233	0.012	0.0129	0.0286	0.2297	0.3952
2	4q-	0.0323	0.0557	0.0185	0.0104	0.0123	0.0222	0.0435	0.0491	0.3875	0.2324
3	7q+	0.2145	0.1836	0.1365	0.1499	0.0858	0.08	0.057	0.0376	0.0303	0.0178
4	8p-	0.0077	0.1366	0.1418	0.0878	0.0634	0.1087	0.2967	0.119	0.0343	0.0032
5	8q+	0.2719	0.1821	0.0608	0.0782	0.1823	0.1154	0.0534	0.027	0.0212	0.0063
6	13q+	0.0174	0.0873	0.1649	0.2302	0.1869	0.1489	0.0831	0.0412	0.0246	0.0115
7	15q-	4e-04	7e-04	0.001	0.0021	0.003	0.0047	0.0057	0.0132	0.0984	0.2945
8	17p-	0.002	0.0102	0.0252	0.0511	0.0746	0.0702	0.1323	0.4954	0.111	0.0233
9	18q-	0.0074	0.0218	0.0824	0.0768	0.156	0.2985	0.1651	0.1439	0.0383	0.0092
10	20q+	0.4089	0.1299	0.1168	0.0618	0.0761	0.0638	0.1051	0.0233	0.0093	0.0033
11	Xq+	0.0359	0.1868	0.2429	0.2365	0.1361	0.0755	0.0452	0.0218	0.0153	0.0033

Table 5.4: Observation number in Habermann's dataset

feature	20q+	1q+	7q+	8q+	7p+	Xq+	13q+	20p+	Xp+	8p-	18q-	16p+	6q+	3q+	5p+	6p+
obs-number	66	46	46	46	45	39	38	36	35	33	32	25	25	24	24	20
feature	19p+	17q+	14q+	16q-	18p-	19q+	4q-	5q+	4p-	10q+	11p+	17p-	18p+	9q+	10p+	16q+
obs-number	19	18	15	15	15	15	15	14	13	12	12	12	12	12	11	11
feature	2q+	12q+	4q+	18q+	2p+	11q+	17p+	5q-	9p-	20p-	22q+	15q+	21q-	8p+	9p+	Yq-
obs-number	11	10	10	9	9	8	8	8	8	7	7	6	6	6	6	6
feature	21q+	3p+	6q-	9q-	11q-	13q-	14q-	3p-	4p+	6p-	Yp-	10q-	12q-	15p+	17q-	22q-
obs-number	5	5	5	5	4	4	4	4	4	4	4	3	3	3	3	3
feature	12p-	Xq-	Yp+	10p-	11p-	16p-	19p-	1p-	21p-	2p-	2q-	3q-	5p-	7p-	7q-	8q-
obs-number	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 5.5: Number of observation of features identified in TCGA study

feature	20q+	1q+	7q+	8q+	7p+	13q+	20p+	8p-	18q-	18p-	19q+	4q-	17p-	12q+	5q-	20p-	8p+	15q-
obs-number	66	46	46	46	45	38	36	33	32	15	15	15	12	10	8	7	6	5

Table 5.6: Number of observational data

features	obs-number
20q+	66
1q+	46
7q+	46
8q+	46
7p+	45
13q+	38
20p+	36
8p-	33
18q-	32
18p-	15
19q+	15
4q-	15
17p-	12
12q+	10
5q-	8
20p-	7
8p+	6
15q-	5
14q-	4
17q-	3
22q-	3
1p-	1

Table 5.7: Nr of obs in H-CBN study

	features	obs-number
1	20q+	66
2	7q+	46
3	8q+	46
4	Xq+	39
5	13q+	38
6	8p-	33
7	18q-	32
8	4q-	15
9	17p-	12
10	15q-	5
11	1p-	1

Table 5.8: Mean number of feature acquired for comparison study

	Chromosome	Mean.of.FA	SD.of.FA
1	15q-	0.852	1.8135
2	1p-	0.7786	1.2865
3	4q-	0.6765	1.0493
4	17p-	0.57	1.0076
5	18q-	0.4479	0.5087
6	8p-	0.4004	0.52
7	13q+	0.3388	0.2875
8	Xq+	0.2624	0.2385
9	7q+	0.2479	0.1373
10	8q+	0.2358	0.2295
11	20q+	0.1897	0.1831

Bibliography

[Andrieu and Roberts, 2009] Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient monte carlo computations. *The Annals of statistics*, 37(2):697–725.

[Beerenwinkel et al., 2007] Beerenwinkel, N., Eriksson, N., and Sturmfels, B. (2007). Conjunctive bayesian networks. *Bernoulli : official journal of the Bernoulli Society for Mathematical Statistics and Probability*, 13(4):893–909.

[Beerenwinkel et al., 2015] Beerenwinkel, N., Schwarz, R., Gerstung, M., and Markowetz, F. (2015). Cancer evolution: Mathematical models and computational inference. *Systematic Biology*, 64(1):e1–e25.

[Brooks and Gelman, 1998] Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455.

[Cancer Genome Atlas, 2012] Cancer Genome Atlas, N. (2012). Comprehensive molecular characterization of human colon and rectal cancer. *Nature*, 487(7407):330–337.

[Desper et al., 1999] Desper, R., Jiang, F., Kallioniemi, O., Moch, H., Papadimitriou, C., and Schäffer, A. (1999). inferring tree models for oncogenesis from comparative genome hybridization data. *Journal of Computational Biology*, 6(1):37–51.

[Diaz-Uriarte, 2018] Diaz-Uriarte, R. (2018). Cancer progression models and fitness landscapes: a many-to-many relationship. *Bioinformatics*, 34(5):836–844.

[Fearon and Vogelstein, 1990] Fearon, E. R. and Vogelstein, B. (1990). A genetic model for colorectal tumorigenesis. *Cell*, 61(5):759–767.

[Gelman, 2013] Gelman, A. (2013). *Bayesian data analysis*. Chapman & Hall/CRC texts in statistical science. CRC Press, Boca Raton, third edition. edition.

[Gelman and Rubin, 1992] Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472.

[Gerstung et al., 2009] Gerstung, M., Baudis, M., Moch, H., and Beerenwinkel, N. (2009). Quantifying cancer progression with conjunctive bayesian networks. *Bioinformatics*, 25(21):2809–2815.

[Gerstung et al., 2011] Gerstung, M., Eriksson, N., Lin, J., Vogelstein, B., and Beerenwinkel, N. (2011). The temporal order of genetic and pathway alterations in tumorigenesis. *PLoS One*, 6(11):e27136–e27136.

[Greenbury et al., 2020] Greenbury, S. F., Barahona, M., and Johnston, I. G. (2020). Hypertraps: Inferring probabilistic patterns of trait acquisition in evolutionary and disease progression pathways. *Cell Syst*, 10(1):39–51 e10.

[Hanahan and Weinberg, 2011] Hanahan, D. and Weinberg, R. (2011). Hallmarks of cancer: The next generation. *Cell*, 144(5):646–674.

[Hanahan and Weinberg, 2000] Hanahan, D. and Weinberg, R. A. (2000). The hallmarks of cancer. *Cell*, 100:57–70.

[Hermsen et al., 2002] Hermsen, M., Postma, C., Baak, J., Weiss, M., Rapallo, A., Sciutto, A., Roemen, G., Arends, J. W., Williams, R., Giaretti, W., De Goeij, A., and Meijer, G. (2002). Colorectal adenoma to carcinoma progression follows multiple pathways of chromosomal instability. *Gastroenterology*, 123(4):1109–19.

[Hjelm et al., 2006] Hjelm, M., Höglund, M., and Lagergren, J. (2006). New probabilistic network models and algorithms for oncogenesis. *Journal of Computational Biology*, 13(4):853–865.

[Höglund et al., 2002] Höglund, M., Gisselsson, D., Hansen, G. B., Säll, T., Mitelman, F., and Nilbert, M. (2002). Dissecting karyotypic patterns in colorectal tumors: Two distinct but overlapping pathways in the adenoma-carcinoma transition. *Cancer Res*, 62(20):5939–5946.

[Johnston and Williams, 2016] Johnston, I. G. and Williams, B. P. (2016). Evolutionary inference across eukaryotes identifies specific pressures favoring mitochondrial gene retention. *Cell Systems*, 2(2):101–111.

[Knutsen et al., 2005] Knutsen, T., Gobu, V., Knaus, R., Padilla-Nash, H., Augustus, M., Strausberg, R. L., Kirsch, I. R., Sirotkin, K., and Ried, T. (2005). The interactive online sky/m-fish & cgh database and the entrez cancer chromosomes search database: Linkage of chromosomal aberrations with the genome sequence. *Genes, Chromosomes and Cancer*, 44(1):52–64.

[Murray and Graham, 2016] Murray, I. and Graham, M. (2016). Pseudo-marginal slice sampling. In Gretton, A. and Robert, C. C., editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *JMLR: W&CP*, pages 911–919, Cadiz, Spain.

[Ried et al., 2019] Ried, T., Meijer, G. A., Harrison, D. J., Grech, G., Franch-Expósito, S., Briffa, R., Carvalho, B., and Camps, J. (2019). The landscape of genomic copy number alterations in colorectal cancer and their consequences on gene expression levels and disease outcome. *Molecular Aspects of Medicine*, 69:48–61.

[Schwartz and Schäffer, 2017] Schwartz, R. and Schäffer, Alejandro, A. (2017). The evolution of tumour phylogenetics: principles and practice. *Nature Reviews Genetics*, 18(4):213–229.

[Sheffer et al., 2009] Sheffer, M., Bacolod, Manny, D., Zuk, O., F. Giardina, S., Pincas, H., Barany, F., Paty, P. B., Gerald, W., Notterman, D. A., Domany, E., and Shenk, T. E. (2009). Association of survival and disease progression with chromosomal instability: A genomic exploration of colorectal cancer. *Proc Natl Acad Sci U S A*, 106(17):7131–7136.

[Szabo and Boucher, 2002] Szabo, A. and Boucher, K. (2002). Estimating an oncogenetic tree when false negatives and positives are present. *Math Biosci*, 176(2):219–236.

[Vogelstein et al., 1988] Vogelstein, B., Fearon, E. R., Hamilton, S. R., Kern, S. E., Preisinger, A. C., Leppert, M., Smits, A. M. M., and Bos, J. L. (1988). Genetic alterations during colorectal-tumor development. *The New England journal of medicine*, 319(9):525–532.

[Wasserman, 2004] Wasserman, L. (2004). *All of Statistics. A Concise Course in Statistical Inference.* Springer, Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo.

Appendix A

A.1 Gelman-Rubin convergence test

A test on the convergence of the posterior output was taken before we applied them to simulate a random walk.

A.1.1 Gelman diagnosis in R

gelman.diag function in CODA package for R was applied for the convergence test. Data applied for gelman.diag function are from reworking of 4 output files, [datafile]-posterior-0-[random number seed]-[length index]-[kernel index].txt. This file contains samples from the posterior distribution that HyperTraPS has learned from the data. Output files contains posterior draws of different parameterizations by applying observational data containing 10 features of copy-number aberrations to run HyperTraPS. We applied for each run a different seed numbers to guarantee that each run started from different position. The comparison of within and between list correlations of posterior draws is the criterion for judgement of convergence of MCMC sampling. Considering on former experience, the algorithm applied 10^6 iterations of MCMC sampling to get list of posterior draws for convergence test.
The output of function gelman.diag in r is PSRF (potential scale reduction factor), and MP-SRF (multivariated PSRF) value. Strict criteria for convergence is Rc (or RSRF) ¡1.1 and flexible criteria is ¡1.2f or all parameters, and should be close to 1. The following table shows

results applying mcmc lists to gelman.diag function in R. To run the program, we need at least mcmc list with 2 different random seed numbers, means started from 2 different points of algorithm. of different combinations of list seed randonm number (s).The idea of the PSRF is that if R is not close to 1 (below 1.1 for example) one may conclude that the tested samples were not from the same distribution (chain might not have been converged yet).

Table A.1: Table of PSRF (or Rc) values of convergence diagnosis

samples	Result	mcmc list
H-samples1	all Rc<1.1	s=1-4,s=1-3,s=1,3,4,s=2-4
H-samples2	all Rc<1.2	s=1-4,s=1-3,s=1,3,4,s=2-4
H-samples3	one Rc>1.2	s=2,3
H-samples4	one Rc>1.2	s=1-3,s=2,3
H-samples5	one Rc>1.1	s=1,2
H-samples6	all Rc<1.1	s=1,2
H-samples7	five Rc>1.2	s=1,3
H-samples8	all Rc<1.1	s=2,4
H-samples9	all Rc<1.1	s=2,4
H-samples10	all Rc<1.1	s=1,4

PSRF: potential scale reduction factor, Rc: a corrected version of PSRF by Brooks and Gelman. Strict criteria for convergence is Rc¡1.1 and flexible criteria is Rc¡1.2 for all parameters. Temporary results for samples from Habermann is shown in table. Symbol ”s” in columns is the random number seed index applied running HyperTraPS with length index 4. Different combination of random number seed index for different samples groups from Habermann’s dataset are applied.

A.1.2 Gelman diagnosis plot in R

Graph show results applied to gelman.plot function to H-samples1. In this study, each list includes 110 parameters. Graph shows the development of shrink factors with increasing number of iterations. shows ho the shrink factor changes as iterations number increase. HyperTraPS algorithm collects value of parameter every 10^3 iterations, therefore the number of iterations labeled on x-axis should be multiplied 10^3. Hence graph shows the number of mcmc iterations up to 10^6. The shrink factor reduces sharply after the iterations increase over $2 * 10^5$ and stay on low level closed to 1. V5 shows different picture that the rang of the upper bound of PSRF seems big. This indicates a large size of variances of parameter. Values runs close to 1.

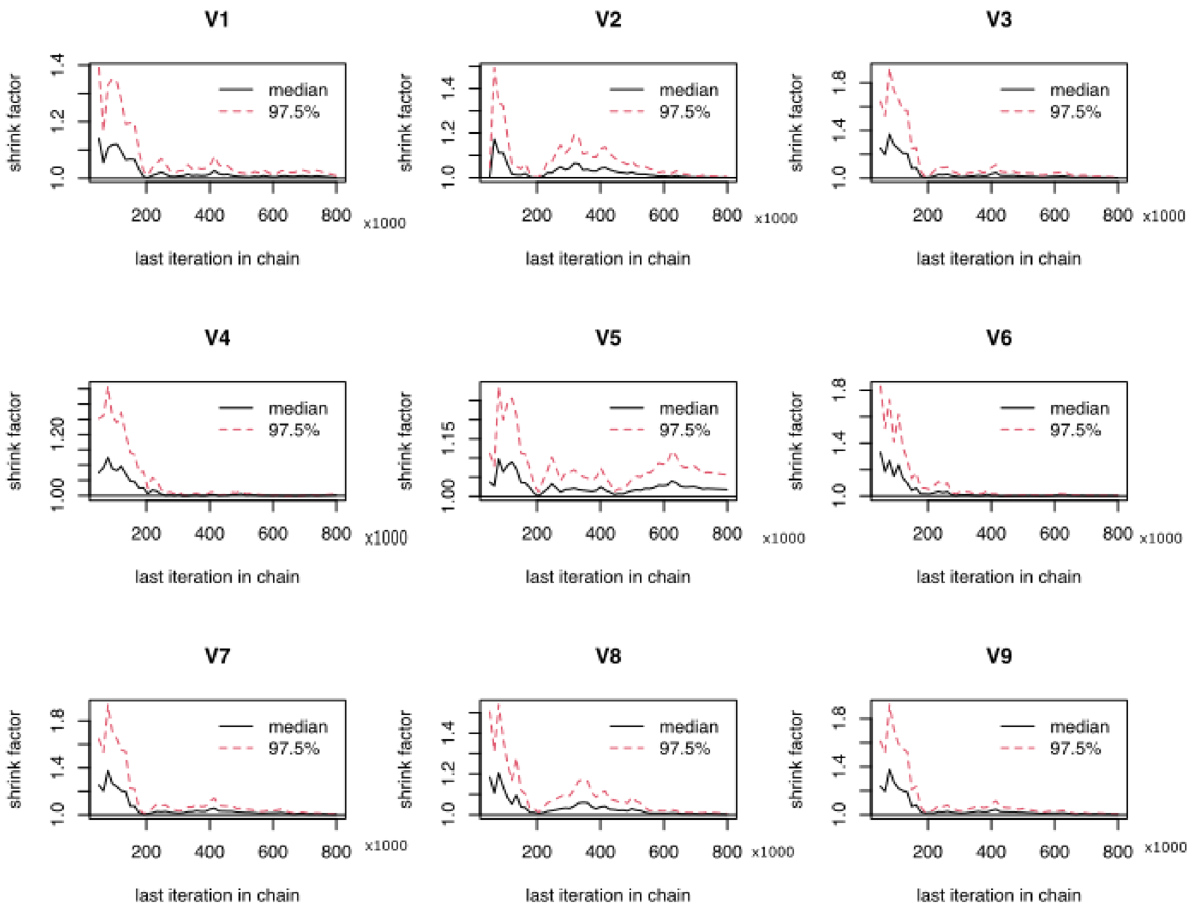
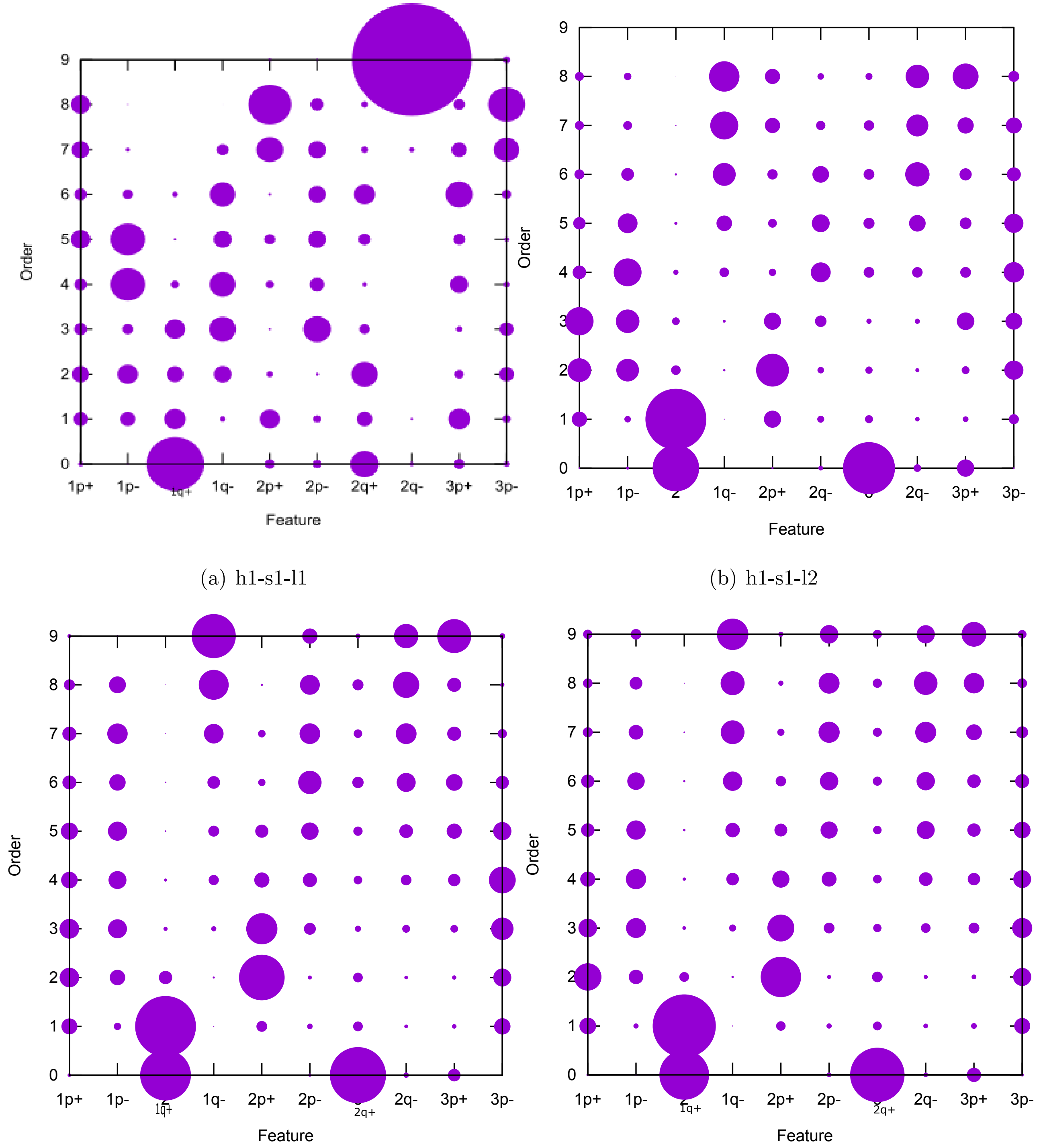


Figure A.1: Gelman convergence diagnosis plot

A.1.3 HyperTraPS heatmap-style output

The following graphs shows heatmap-style output after running HyperTraPS. Dataset applied to run model are the first 10 features (1p+, 1p-, 1q+, 1q-, 2p+, 2p-, 2q+, 2q-, 3p+, 3p-). Figure A.2(a) A.2(b) A.2(c) A.3(a) are output heatmap style on the feature acquired at orderings after running model with different number of iterations. l=1,2,3,4, corresponds the number of iterations : $103, 10^4, 10^5, 10^6$. What noted is the feature 1q- that there is no observation of this feature in the Habermann dataset. Nevertheless, model shows also acquisition of such a feature.

Figure A.3(a) A.3(b) show the feature acquired at different ordering after convergence (iterations 10^6, l4) and different random seed numbers, s=2, s=4. Different random seed number s means runs started from different places. Figures show after convergence, the pattern of feature acquired ordering are identical.



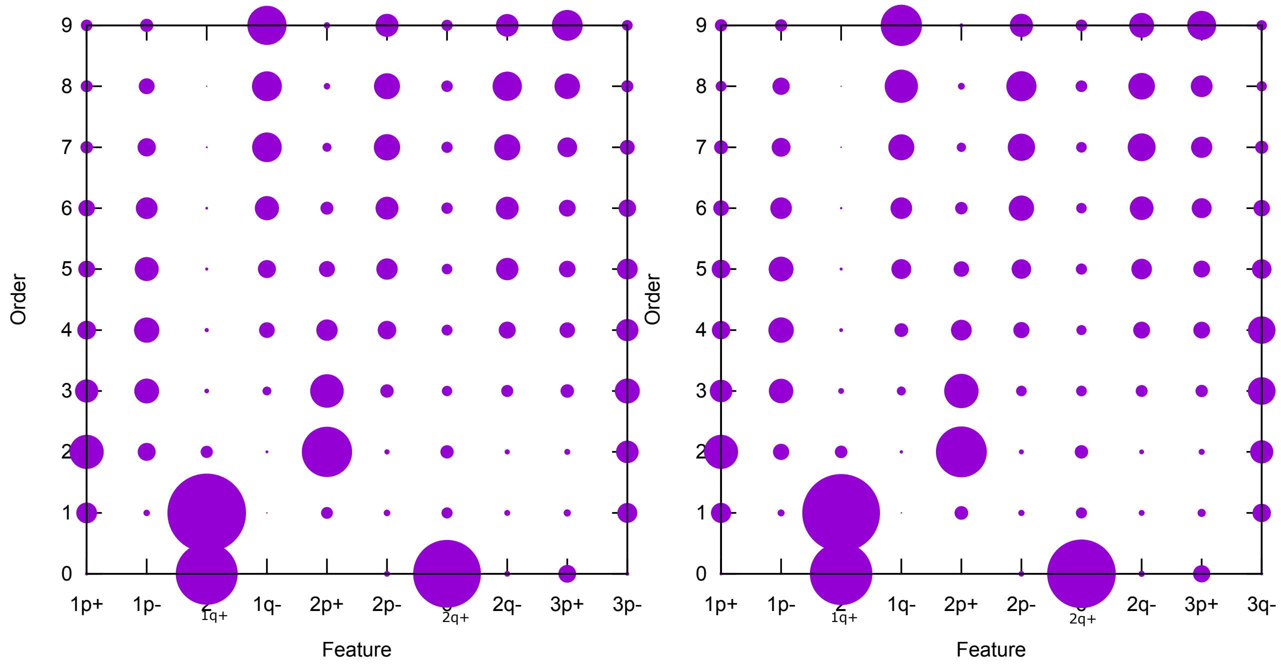
(a) h1-s1-l1

(b) h1-s1-l2

(c) h1-s1-l3

(d) h1-s2-l4

Figure A.2: H1 samples use l=1-4



(a) h1-s2-l4

(b) h1-s4-l4

Figure A.3: H1 samples use s2,s4

Appendix B

B.1 Python code: create input data

Python script to parse some SKY records (J.Habermann.esi). Using regular expressions in Python with the re module to extract get ID lines contains both ”SkyCase” and chromosomal aberrations information (”CGHFrag”).Python script also transforms chromosomal aberrations information to build a binary set of traits for samples from J.Habermann.Binary set of traits is piped to a matrix and stored as a csv file. An intermediate file records total number of each chromosomal aberrations that Habermann dataset contains.

Listing B.1: Python code 1: extract SkyCase-CGHFrag from esi file

```python
import re
#Dataset from J.Habermann contains only cases from human
# Use Regular Expression (RE) pull out all ID lines (SkyCase) or lines containing aberration details to a temporary file (CGHFrag)

outFile1 = open("H_r1.tmp","w+")
fileD  = open("J.Habermann.esi", "r")
pattern='SkyCase|CGHFrag'
for line in fileD:
    result =re.match(pattern, line)
    if result:
        outFile1.write(line)
outFile1.close()
fileD.close()
```

If line is an ID line (contains ”SkyCase”), print current filename and ID, otherwise use RE to find the start and the end region of chromosome. Construct a dictionary and assign ”+” to chromosome if marked with ”gain” and ”-” if with ”loss”.

Listing B.2: Python code 2: list features

```
outFile2= open("H_r2.tmp","w+")
GLdist = {"Gain":"+", "Loss":"-"}
with open('H_r1.tmp','r') as fil:
    lines=fil.readlines()
    # if line is an ID line (contains "SkyCase") print current filename and ID
    for line in lines:
        lineList = []
        lineList2 = []
        if 'SkyCase'in line:
                outFile2.writelines(line)
# use RE to find the chromosome number
         elif 'CGHFrag'in line:
             firstNumber = re.search('[0-9]+|[X-Y]',line)
             N = firstNumber.group(0)
             lineList.append(N)
# use RE find the start and end chromosome
            pq = re.findall('p|q',line)
            if pq[0] == pq[1]:
                lineList.append(pq[0])
            # if both chromosomes the same, write one line with this chromosome
                result1=re.search('Gain|Loss',line)
                if result1:
                    lineList.append(GLdist[result1.group(0)])
                strj = ""
                outFile2.writelines(strj.join(lineList)+"\n")
            else:
                lineList2 = lineList + [pq[1]]
                lineList.append(pq[0])
              #otherwise write two lines for both
                result1=re.search('Gain|Loss',line)
                if result1:
                    lineList.append(GLdist[result1.group(0)])
                    lineList2.append(GLdist[result1.group(0)])
                strj = ""
                outFile2.writelines(strj.join(lineList)+"\n")
                outFile2.writelines(strj.join(lineList2)+"\n")
# also assign "+" if chromosome marked with "gain" and "-" if with "loss"
#e.g.  3 0 q11.2 q29 cghGain -> "3 q+" ; 18 0 pter q23 cghLoss ->"18p-" and "18q-"

outFile2.close()
```

Construct a file to enable count only instances of chromosome aberrations. File contains no line ID and only the number of chromosome, sign of start region, of end region and of gain/loss.

Listing B.3: Python code 3: create file for count features

```
outFile3= open("H_r2_count.tmp","w+")
GLdist = {"Gain":"+", "Loss":"-"}
with open('H_r1.tmp','r') as fil:
    lines=fil.readlines()
    for line in lines:
        lineList = []
        lineList2 = []
        if 'CGHFrag'in line:
            firstNumber = re.search('[0-9]+|[X-Y]',line)
            N = firstNumber.group(0)
            lineList.append(N)
            pq = re.findall('p|q',line)
            if pq[0] == pq[1]:
                lineList.append(pq[0])

                result1=re.search('Gain|Loss',line)
                if result1:
                    lineList.append(GLdist[result1.group(0)])
                strj = ""
                outFile3.writelines(strj.join(lineList)+"\n")
            else:
                lineList2 = lineList + [pq[1]]
                lineList.append(pq[0])

                result1=re.search('Gain|Loss',line)
                if result1:
                    lineList.append(GLdist[result1.group(0)])
                    lineList2.append(GLdist[result1.group(0)])

                strj = ""
                outFile3.writelines(strj.join(lineList)+"\n")
                outFile3.writelines(strj.join(lineList2)+"\n")
outFile3.close()
```

Count overall instances of aberration and write to the output file.

Listing B.4: Python code 4: create feature list with tag ”+”,”-”

```
outFile4= open("H_r3.tmp","w+")
with open('H_r2_count.tmp','r') as fil:
    lines=fil.readlines()
    linetable={}
    for line in lines:
        linelist=line.split('\n')
        linestr="".join(linelist)
        countN=linetable.get(linestr,0)
        linetable[linestr]=countN+1

for item in linetable:
    value=linetable[item]
    outFile4.write('%-5s%3s\n' % (item,value))
outFile4.close()
```

Get overall instance of chromosomal aberrations for each case and write down them on document. Such like for sample number 4: 04 3q+ 7p+ 7q+ 8p- 8q+ 9p+ 9q+ 10p+ 10p+ 10p+ 10p+ 10q+ 11p+ 11p+ 11q+ 13q+ 14q+ 16p+ 17p- 17q+ 17q- 18p+ 18q- 20p+ 20q+ 21q- 22q+ Xp+ Xq+ Yq+.

Listing B.5: Python code 5: get overall features for each case

```
outFile5=open("H_r4.tmp","w+")
with open('H_r2.tmp','r') as fil:

    lines=fil.readlines()
    lineList = []

    for line in lines:
        if ’SkyCase’ in line:
            if lineList != []:
                outFile5.writelines(’’.join(lineList)+"\n")
            lineList = []
            firstNumber = re.search(’[0-9]+’,line)
            N = firstNumber.group(0)
            lineList.append(N+’ ’)
        else:
            lineList.append(line[:-1]+’ ’)

outFile5.close()
```

Construct a matrix indicating instances of chromosome aberrations, as ”1” if exists, otherwise ”0” for each sample. Use sample ID to label rows of matrix.

Listing B.6: Python code 6: create matrix indicate instance of all aberrations

```
import numpy as np

bigDict = {}
colIndex = 0
for seqN in ([s for s in range(1,23)] + [’X’]+[’Y’]):
    for pq in [’p’,’q’]:
        for PN in [’+’, ’-’]:
            bigDict[str(seqN) + pq + PN] = colIndex
            colIndex += 1

outFile6 = open(’H_outMatrix.csv’, ’w+’)
outFile6.write("ID, "+’, ’.join(bigDict.keys())+"\n")

with open(’H_r4.tmp’,’r’) as fil:
    lines=fil.readlines()
    for line in lines:

        bigArr = np.zeros((colIndex,),dtype=int)
        splitObj = line.split()
        ID = splitObj[0]
        splitObj.pop(0)
        bigArr[[bigDict[rec] for rec in splitObj]] = 1
        outFile6.write(ID+", "+str(bigArr.tolist()).strip(’[]’)+"\n")
```

B.2 R code

Create input dataset for cross-sectional independent data to run HyperTraPS

Listing B.7: Rcode: create cross-sectional input data

```
library(dplyr)
#the output file of chromosomal aberration of traits for Habermann
   ↪ dataset is converted to a text tile
```

```
habermann<-read.csv("H_outMatrix.csv",header=TRUE)
habermann<-data.frame(habermann)
habermann<-habermann[,-1]
habermann<-as.matrix(habermann)

#create an empty matrix without any row, so we can use rbind to add %new rows into the empty matrix
zerorow<-rep(0,96)
habermann_ny <- habermann[rep(1:nrow(habermann), each = 2), ] #double rows of habermann
habermann_ny[1:nrow(habermann_ny) %% 2 == 0, ] <- zerorow #set the even number row as zerorow
habermann_ny<-rbind(zerorow,habermann_ny) #add a zerorow to habermann_ny as first row
habermann_ny<-habermann_ny[-nrow(habermann_ny),] #extract the last zerorow in matrix
write.table(habermann_ny, "habermann-cross-samples-data.txt",row.names=FALSE, col.names=FALSE)

Create various dataset for running HyperTraPS

\begin{lstlisting}[caption={R code 2},label=code8]
habermann_cross_samples<-read.table("habermann-cross-samples-data.txt",header=FALSE)
habermann_cross_samples<-data.frame(habermann_cross_samples)
habermann_cross_samples_1<-habermann_cross_samples[1:124,1:10]
habermann_cross_samples_2<-habermann_cross_samples[1:124,11:20]
habermann_cross_samples_3<-habermann_cross_samples[1:124,21:30]
habermann_cross_samples_4<-habermann_cross_samples[1:124,31:40]
habermann_cross_samples_5<-habermann_cross_samples[1:124,41:50]
habermann_cross_samples_6<-habermann_cross_samples[1:124,51:60]
habermann_cross_samples_7<-habermann_cross_samples[1:124,61:70]
habermann_cross_samples_8<-habermann_cross_samples[1:124,71:80]
habermann_cross_samples_9<-habermann_cross_samples[1:124,81:90]
habermann_cross_samples_10<-habermann_cross_samples[1:124,91:96]
```

Apply R code on at least 2 output files [datafile]-posterior-0-[random number seed]-[length index]-[kernel index].txt with different random number seed to do G-R convergence test. Write down the case with PSRF¿1.1 in a file.

Listing B.8: Rcode: G-R test

```
install.packages("conda")
checkpackages=function(package){
  if (!package %in% installed.packages())
    install.packages(package)
}
listpackage=c("coda","lattice")
lapply(listpackage,checkpackages)

library("coda","lattice")
s114<-as.matrix(read.table("habermann-cross-samples-1.txt-posterior-0-1-4-5.txt"))
s114.mcmc<-as.mcmc(s114, start=1, end=length(s114))
s214<-as.matrix(read.table("habermann-cross-samples-1.txt-posterior-0-2-4-5.txt"))
s214.mcmc<-as.mcmc(s214, start=1, end=length(s214))
s314<-as.matrix(read.table("habermann-cross-samples-1.txt-posterior-0-3-4-5.txt"))
s314.mcmc<-as.mcmc(s314, start=1, end=length(s314))
s414<-as.matrix(read.table("habermann-cross-samples-1.txt-posterior-0-4-4-5.txt"))
s414.mcmc<-as.mcmc(s414, start=1, end=length(s414))
seed.14.samples1.all <- mcmc.list(s114.mcmc,s214.mcmc,s314.mcmc,s414.mcmc)
```

```

gelmandiag1<-gelman.diag(seed.l4.samples1.all, confidence = 0.95, transform=FALSE)
g1<-gelmandiag1$psrf>1.1

for (item in names(gelmandiag1)) write.csv( file=paste("gelmandiag1 List item",item), gelmandiag1[[item]] )
dat<-read.table("gelmandiag1 List item psrf",fill = TRUE , header = FALSE)
```

R codes listed in this subsection were written to infer the dynamic of features acquisition from the output file containing the estimand of posterior.

Listing B.9: Rcode: count feature number Haberamann dataset

```
library(dplyr)
dat<-read.table("H_r3.tmp")
datm<-data.frame(dat)
sum(datm$V2)
colnames(datm)<-c("features","obs-number")
x<-seq(1:96)
featuresL<-c("1p+", "1p-", "1q+", "1q-", "2p+", "2p-", "2q+", "2q-", "3p+", "3p-",
             "3q+", "3q-", "4p+", "4p-", "4q+", "4q-", "5p+", "5p-", "5q+", "5q-", "6p+",
             "6p-", "6q+", "6q-", "7p+", "7p-", "7q+", "7q-", "8p+", "8p-", "8q+", "8q-",
             "9p+", "9p-", "9q+", "9q-", "10p+", "10p-", "10q+", "10q-", "11p+", "11p-",
             "11q+", "11q-", "12p+", "12p-", "12q+", "12q-", "13p+", "13p-", "13q+", "13q-",
             "14p+", "14p-", "14q+", "14q-", "15p+", "15p-", "15q+", "15q-", "16p+", "16p-",
             "16q+", "16q-", "17p+", "17p-", "17q+", "17q-", "18p+", "18p-", "18q+", "18q-",
             "19p+", "19p-", "19q+", "19q-", "20p+", "20p-", "20q+", "20q-", "21p+", "21p-",
             "21q+", "21q-", "22p+", "22p-", "22q+", "22q-", "Xp+", "Xp-", "Xq+", "Xq-", "Yp+",
             "Yp-", "Yq+", "Yq-")
m<-data.frame(x,featuresL)
colnames(m)<-c("orig-feature","features")
mc<-merge(m, datm, by="features")
mc_end<-t(mc[order(-mc$‘obs-number‘),-2])
colnames(mc_end)<-NULL
write.csv(mc_end,"H-features obs-numbers-a.csv")
```

Find the mean and SD of mean of feature acquired in the simulated random walk applied to 22traits in Habermann dataset,id in TCGA study. R code applied on output file ”habermann-cross-samples-cancer1-s3-l4” after running HyperTraPS.

Listing B.10: Rcode: mean and SD of feature acquisition

```
library(dplyr)
dat_cancer1<-read.table("habermann-cross-samples-cancer1-s3-l4")
# x is the output file "posteriors.ce":
#x[,1]: ordering index, begins with 0.
```

```r
# x[,5]: the posterior probability of weighted edge
#Feature get 1 ordering forwarding one step, this is also times feature encountered
#mean (acquired order/feature): the product of (x[,1]:index of ordering) and ( x[,5]: the posterior probability of weighted edg)

getorder<-function(x){
  x[,5]<-x[,1]*x[,5]
  return(x)
}

dat_cancer1_tmp<-getorder(dat_cancer1)
#extract feature name and order acquired on each ordering
#column 5 is the order acquired on each ordering and column 3 is the running position of each feature
d_cancer1<-dat_cancer1_tmp[,c(3,5)]
#to use tapply function to calculate  average acquired order of each feature.
#data must be a dataframe form and we give name to columns for calculation.
#we give column name  "feature" to column of feature label, and "p" to column of posterior probability of weighted edge
colnames(d_cancer1)[1]<-"feature"
colnames(d_cancer1)[2]<-"p"

#calculate mean order acquired, SD of order acquired and CV=mean/SD of order acquired
#cancer1sum<-round(tapply(d_cancer1$p, d_cancer1$feature, sum),digits=4)

mean<-round(tapply(d_cancer1$p, d_cancer1$feature, mean),digits=4) #cancer1
sd<-round(tapply(d_cancer1$p, d_cancer1$feature, sd),digits=4)#cancer1

#Additional index: index included in result file
#1. run_feature: the original feature index running HyperTraPS
#2. orig_feature: the index of feature according to the assemble indexing all 23 human chromosome from 1 to 96
#3. chromosome: index of the feature chromosomal position with tag "+ " and "-" indicating gain and loss of chromosome.
feature_index<-sort(dat_cancer1[,3][1:22]) #cancer1
orig_feature<-c(paste0("feature", c(2,3,16,20,25,27,29,30,31,47,51,56,60,66,68,70,72,75,77,78,79,88))) #cancer1
featuresL<-c("1p+", "1p-", "1q+", "1q-", "2p+", "2p-", "2q+", "2q-", "3p+", "3p-", "3q+", "3q-", "4p+", "4p-", "4q+",
             "4q-", "5p+", "5p-", "5q+", "5q-", "6p+", "6p-", "6q+", "6q-", "7p+", "7p-", "7q+", "7q-", "8p+", "8p-",
             "8q+", "8q-", "9p+", "9p-", "9q+", "9q-", "10p+", "10p-", "10q+", "10q-", "11p+", "11p-", "11q+", "11q-",
             "12p+", "12p-", "12q+", "12q-", "13p+", "13p-", "13q+", "13q-", "14p+", "14p-", "14q+", "14q-", "15p+",
             "15p-", "15q+", "15q-", "16p+", "16p-", "16q+", "16q-", "17p+", "17p-", "17q+", "17q-", "18p+", "18p-",
             "18q+", "18q-", "19p+", "19p-", "19q+", "19q-", "20p+", "20p-", "20q+", "20q-", "21p+", "21p-", "21q+",
             "21q-", "22p+", "22p-", "22q+", "22q-", "Xp+", "Xp-", "Xq+", "Xq-", "Yp+", "Yp-", "Yq+", "Yq-")
chromosome<-featuresL[c(2,3,16,20,25,27,29,30,
31,47,51,56,60,66,68,70,72,75,77,78,79,88)] #cancer1
# combine all additional index and mean, sd and cv of order acquired together
# features are sorted after size of mean ascending
H_cancer1_order<-cbind(chromosome,feature_index,mean,sd)
```

```
H_cancer1_order<-as.data.frame(H_cancer1_order)
H_cancer1_order_s<-H_cancer1_order[order(-mean),]
#naming
row.names(H_cancer1_order_s)<-NULL
colnames(H_cancer1_order_s)<-c("Chromosome","Feature index","Mean of FA","SD of FA") #cancer1
write.csv(H_cancer1_order_s,"Mean number of feature acquired.csv")
```

Transition probability at different orderings: this file shows the intensity of edge on all transition steps.

Listing B.11: Rcode: create table trans. probability

```
dat_cancer1<-read.table("habermann-cross-samples-cancer1-s3-l4")
#The posterior probability of feature on each ordering: calculated by multiply original feature index, start with 1.
# x is the output file "posteriors.ce": x[,3]: original feature index, begins with 0.
featuresq<-function(x){
  x[,3] <- x[,3] + 1
  return(x)
}
dat_cancer1_tmp<-featuresq(dat_cancer1)

#extract transition probability at different ordering
#posterior.ce" [,5]: posterior probability on each ordering
#x[,3]:original feature index+1 because program can begin from 1 not 0
postcancer1<-function(x){
  m<-matrix(ncol=22,nrow=0)
  for (i in 1:22){
    v<-x[x$V3==i,5]
    m<-rbind(m,v)
  }
  return(m)
}
dat_cancer1_posterior<-data.frame(round(postcancer1(dat_cancer1_tmp),digits=4))

#feature_index: index included in result file, sorted from small to large number
#chromosome: index of the feature chromosomal position with tag "+ " and "-" indicating gain and loss of chromosome.
featuresL<-c("1p+", "1p-", "1q+", "1q-", "2p+", "2p-", "2q+", "2q-", "3p+", "3p-", "3q+", "3q-", "4p+", "4p-", "4q+",
             "4q-", "5p+", "5p-", "5q+", "5q-", "6p+", "6p-", "6q+", "6q-", "7p+", "7p-", "7q+", "7q-", "8p+", "8p-",
             "8q+", "8q-", "9p+", "9p-", "9q+", "9q-", "10p+", "10p-", "10q+", "10q-", "11p+", "11p-", "11q+", "11q-",
             "12p+", "12p-", "12q+", "12q-", "13p+", "13p-", "13q+", "13q-", "14p+", "14p-", "14q+", "14q-", "15p+",
             "15p-", "15q+", "15q-", "16p+", "16p-", "16q+", "16q-", "17p+", "17p-", "17q+", "17q-", "18p+", "18p-",
             "18q+", "18q-", "19p+", "19p-", "19q+", "19q-", "20p+", "20p-", "20q+", "20q-", "21p+", "21p-", "21q+",
             "21q-", "22p+", "22p-", "22q+", "22q-", "Xp+", "Xp-", "Xq+", "Xq-", "Yp+", "Yp-", "Yq+", "Yq-")
chromosome<-featuresL[c(2,3,16,20,25,27,29,30,31,47,51,56,60,66,68,70,72,75,77,78,79,8 #cancer1
feature_index<-sort(dat_cancer1[,3][1:22]) #cancer1
#add row and column name: row name of file are the index of feature in the human chromosomal arm sequence, start from 1 to 96
```

```
#column name: the ordering
dat_cancer1_posterior_n<-cbind(chromosome,feature_index,dat_cancer1_posterior) #cancer1
names(dat_cancer1_posterior_n)<-c("Chromosome","Feature index", c(paste0("ordering", 0:21)))
row.names(dat_cancer1_posterior_n)<-NULL
write.csv(dat_cancer1_posterior_n,"Transition probability of features at different orderings.csv")
```

R code to find the most, the second most likely transition pathways

Listing B.12: Rcode: find likely trans. pathways

```
install.packages("Rfast")
library(Rfast)
dat_cancer1_posterior<-read.csv("Transition probability of features at different orderings.csv", sep=",")#cancer1
row.names(dat_cancer1_posterior)<-dat_cancer1_posterior[,2]
dat_cancer1_posterior<-dat_cancer1_posterior[-18,-(1:3)]#delete 19q+

mat<-data.frame(dat_cancer1_posterior) #as data.frame, cancer1

#The most likely feature acquired ordering cancer1
col_idx<-array()
idx_name<-array()
maxp<-array()
next_sample=1
for (i in 1:20){
idx<-which.max(mat[,i])
idxname<-rownames(mat)[idx]
p<-mat[idx,i]
mat[idxname,]<--1 #set value=-1 to those features that are already acquired
col_idx[[next_sample]]<-idx
idx_name[[next_sample]]<-idxname
maxp[[next_sample]]<-p
next_sample=next_sample+1
}
# The second most likely for cancer 1

idx_name<-array()
mat2<-array()
col_idx<-array()
next_sample=1
for (i in 1:20){
  value<-Rfast::nth(mat[,i], 2, descending = T)
  idx<-which(mat[,i] ==value, arr.ind = T)
  idxname<-rownames(mat)[idx]
  p<-mat[idx,i]
  mat[idxname, ]<--1 #set value=-1 to those features that are already acquired
  mat2[[next_sample]]<-p
  col_idx[[next_sample]]<-idx
  idx_name[[next_sample]]<-idxname
  next_sample=next_sample+1
}
setdiff(rownames(mat),idx_name)
idx_name[21]<-setdiff(rownames(mat),idx_name)
mat2[21]<-max(mat[,21])

# Combine the name vector
```

```
#The most likely pathway
facqorder<-rbind(idx_name,maxp)
colnames(facqorder)<-c(paste0("ordering", 0:20)) #cancer 1
row.names(facqorder)<-c("Chromosome", "Probability")
write.csv(facqorder,"The most likely transition pathway_c1a.csv")
#The second most likely pathway
facqorder<-cbind(idx_name, mat2)
row.names(facqorder)<-c(paste0("ordering",0:20) )#cancer 1
colnames(facqorder)<-c("Chromosome", "Probability")
write.csv(facqorder,"The second most likely transition pathway_c1.csv")
```