Supplementary Methods

Gut microbiome analysis as a tool towards targeted non-invasive biomarkers for early hepatocellular carcinoma

Running title: Gut microbiome for early hepatocellular carcinoma

Zhigang Ren[1,3,2†], Ang Li[2,4,3†], Jianwen Jiang[1,4,9†], Lin Zhou[1,4†], Zujiang Yu[3,2†], Haifeng Lu[4], Haiyang Xie[1,4], Xiaolong Chen[3,2], Li Shao[4], Ruiqing Zhang[5,8], Shaoyan Xu[1], Hua Zhang[4], Guangying Cui[3,2], Xinhua Chen[1,4], Ranran Sun[3,2], Hao Wen[8], Jan Lerut[6], Quancheng Kan[7]*, Lanjuan Li[4]*, and Shusen Zheng[1,4,10]*

[1] Department of Hepatobiliary and Pancreatic Surgery, the First Affiliated Hospital, School of Medicine, Zhejiang University; Key Laboratory of Combined Multi-organ Transplantation, Ministry of Public Health, Hangzhou 310003, China.

[2] Gene Hospital of Henan Province; Precision Medicine Center, the First Affiliated Hospital of Zhengzhou University, Zhengzhou 450052, China.

[3] Department of Infectious Diseases, the First Affiliated Hospital of Zhengzhou University, Zhengzhou 450052, China.

[4] State Key Laboratory for Diagnosis and Treatment of Infectious Disease; Collaborative Innovation Center for Diagnosis and Treatment of Infectious Diseases, Zhejiang University, Hangzhou 310003, China.

[5] Hepatobiliary and Hydatid Department, Digestive and Vascular Surgery Centre, Xinjiang Key Laboratory of Echinococcosis, the First Affiliated Hospital of Xinjiang Medical University, Urumqi, Xinjiang 830054, China.

[6] Starzl Unit Abdominal Transplantation, University Hospitals Saint Luc, Université

catholique Louvain, UCL Brussels, Belgium.

[7] Department of Pharmacy, the First Affiliated Hospital of Zhengzhou University, Zhengzhou 450052, China.

[8] State Key Laboratory of Pathogenesis, Prevention and Treatment of High Incidence Diseases in Central Asia, Xinjiang Medical University, Urumqi, Xinjiang 830054, China.

[9] Health Management Center, First Affiliated Hospital, School of Medicine, Zhejiang University, Hangzhou 310003, China.

[10] Department of Hepatobiliary and Pancreatic Surgery, Shulan (Hangzhou) Hospital, Hangzhou 310022, China.

[†]These authors contributed equally to this work.

Supplementary Methods

Inclusion and exclusion criteria of participants

HCC or liver cirrhosis was diagnosed according to the international guidelines by comprehensive integration of imaging, clinical symptoms and physical signs, laboratory tests and medical history. The diagnosis was confirmed by histopathological examination of specimens from surgical resection or percutaneous ultrasound-guided liver needle core biopsy, being the ‘gold standard’ for HCC or cirrhosis diagnosis. Tumor stage was determined according to the Barcelona Clinic Liver Cancer (BCLC) stage [1 2]. HCC patients with or without cirrhosis were screened and confirmed. Exclusion criteria were as follows: (a) intrahepatic cholangiocarcinoma; (b) prior anticancer treatment; (c) presence of other diseases such as hypertension, diabetes and metabolic disease; and (d) participants missing clinical information. Tumor differentiation was graded following Edmondson [3] and liver function was assessed following the Child-Turcotte-Pugh.

The control group consisted of 131 healthy volunteers who visited our hospital for their annual physical examination. The inclusion criteria for healthy volunteers referred to our previous study [4]. In all healthy controls, physical examination, liver biochemistry, routine examination of blood, urine and stools, serological tests (including the detection of hepatitis B surface antigen, hepatitis C virus antibody, Treponema pallidum antibody, human immunodeficiency virus antibody), liver function, renal function, electrolyte, liver ultrasound, electrocardiogram and chest X-ray results were in the normal range. Exclusion criteria for healthy volunteers included hypertension, diabetes, obesity, metabolic syndrome, irritable bowel syndrome (IBD), nonalcoholic fatty liver disease, coeliac disease and liver cirrhosis. Individuals who received antibiotics and/or probiotics within 8 weeks before enrolment were also excluded.

Clinical images including computed tomography (CT) scan and enhanced CT were collected for healthy control, early cirrhosis patients, early HCC with cirrhosis patients, advanced HCC with cirrhosis patients, early HCC patients and advanced HCC patients. Also, the histopathology staining was conducted, and the representative images were selected for healthy control, liver cirrhosis and HCC patients.

Human fecal sample collection and DNA extraction

Each individual provided a fresh stool sample at 7:00-8:30 am; this was delivered immediately from our hospital to the laboratory in an ice bag using insulating polystyrene foam containers. In the laboratory, the sample was divided into five aliquots of 200 mg and immediately stored at -80 ℃. The sample that stayed in room temperature more than 2 hours was discarded. A frozen aliquot (200 mg) of each fecal sample was processed by phenol trichloromethane DNA extraction using a bead beater to mechanically disrupt cells, followed by phenol–chloroform extraction, as we previously described [5 6]. DNA was further purified using the Quick gel extraction kit (Qiagen, Germany) according to the manufacturer's instructions. DNA concentration was measured by NanoDrop (Thermo Scientific), and its molecular size was estimated by agarose gel electrophoresis.

PCR amplification and MiSeq sequencing

The extracted DNA samples were amplified with a set of primers targeting the hypervariable V3-V5 region (338F/806R) of the 16S rRNA gene. The forward primer is 5'-ACTCCTACGGGAGGCAGCA-3' and the reverse primer is 5'-GGACTACHVGGGTWTCTAAT-3'. Barcode and adapter were incorporated between the adapter and the forward primers. The PCR amplification was performed in a 20μl reaction system containing 4μl 5×Fastpfu Buffer, 2μl 2.5mM dNTPs, 0.4μl Forward Primer (5μM), 0.4μl Reverse Primer (5μM), 0.4μl TransStart Fastpfu DNA

Polymerase (TransGen Biotech, Beijing, China), and 10ng Template DNA. The PCR was conducted in a PCR machine (ABI GeneAmp® 9700) under the following conditions: 95 ℃ for 2 min; 30 cycles of 95 ℃ for 30 s, 55 ℃ for 30 s, 72 ℃ for 30 s, and completed with a final extension at 72 ℃ for 5 min. PCR products were detected on a 2 % (w/v) agarose gel, and the band was extracted and purified using the AxyPrepDNA Gel (Axygen, CA, USA) and PCR Clean-up System. The purified PCR product for each sample was mixed. DNA libraries were constructed according to the manufacturer's instructions, and the sequencing was performed on the IlluminaMiSeq platform by Shanghai Itechgene Technology Co. Ltd., China. The raw Illumina read data for all samples have been deposited in the European Bioinformatics Institute European Nucleotide Archive database under the accession number **PRJEB8708.**

Sequence data process

According to the specific barcodes, the filtered reads were assigned into different samples, and then the barcodes and primers were trimmed off. The amplified reads were processed with following steps: (a) pair end sequenced reads of each library were overlapped by FLASH version 1.2.10 [7] with default parameters. (b) a custom per program was used to perform more specific quality control of overlapped reads generated by FLASH: 1) No ambiguous bases (N) were allowed in reads; 2) No more than 5 mismatches were allowed in overlap region; 3) No mismatches were allowed in barcode/primer region. (c) reads were de-multiplexed and assigned into different samples according to barcodes; (d) chimeric sequences were detected and removed with UCHIME version 4.2.40 [8] with 16S "golden standard" database provided by Broad Institute as reference (version microbiome util-r20110519, http://drive5.com/uchime/gold.fa) to match Operational Taxonomy Units (OTUs).

OTUs clustering and taxonomy annotation

We randomly chose reads from all samples with equal number, and then OTUs were binned by UPARSE pipeline [9] with following steps: (a) abundant sequences and singletons were firstly removed; (b) unique sequences were binned into OTUs with command "usearch-cluster_otus"; (c) randomly chosen sequences were aligned against OTU sequences with command "usearch-usearch_global-id 0.97", the identity threshold was set as 0.97, and then OTU composition table was created.

Bacterial diversity and taxonomic analysis

The weighted unifrac distances was calculated with phyloseq package[10] with following command: "Unifrac (X1, weighted=T, normalized=T, fast=T)"; for the unweighted unifrac distances, parameter is "weighted=F". X1 is composition of sequences table and phylogenetic tree. Jensen-Shannon distance was calculated with a custom R program function provided by EBML (http://enterotype.embl.de/enterotypes.html#dm). Spearman coefficient distance was calculated with "as.dist (1-cor (dat), method = "spearman")", data is OTU composition table.

Phylogenetic tree was calculated with three steps: (1) Sequences were aligned by using MUSLE; (2) Fast Tree MP was used to calculate unrooted phylogenetic tree with generalized time-reversible (gtr) model; (3) A custom perl script provided by Microbes Online (reroot.pl, www.microbesonline.org/programmers.html) was used to re-root the phylogenetic tree.

REFERENCES

1. Llovet JM, Bru C, Bruix J. Prognosis of hepatocellular carcinoma: the BCLC staging classification. Semin Liver Dis. 1999;19(3):329-38.

2. Bruix J, Llovet JM. Major achievements in hepatocellular carcinoma. Lancet. 2009;373(9664):614-6.

3. Wittekind C. [Pitfalls in the classification of liver tumors]. Pathologe. 2006;27(4):289-93.

4. Qin N, Yang F, Li A, Prifti E, Chen Y, Shao L, et al. Alterations of the human gut microbiome in liver cirrhosis. Nature. 2014;513(7516):59-64.

5. Ren Z, Jiang J, Lu H, Chen X, He Y, Zhang H, et al. Intestinal microbial variation may predict early acute rejection after liver transplantation in rats. Transplantation. 2014;98(8):844-52.

6. Chen Y, Yang F, Lu H, Wang B, Lei D, Wang Y, et al. Characterization of fecal microbial communities in patients with liver cirrhosis. Hepatology. 2011;54(2):562-72.

7. Magoc T, Salzberg SL. FLASH: fast length adjustment of short reads to improve genome assemblies. Bioinformatics. 2011;27(21):2957-63.

8. Edgar RC, Haas BJ, Clemente JC, Quince C, Knight R. UCHIME improves sensitivity and speed of chimera detection. Bioinformatics. 2011;27(16):2194-200.

9. Edgar RC. UPARSE: highly accurate OTU sequences from microbial amplicon reads. Nat Methods. 2013;10(10):996-8.

10. McMurdie PJ, Holmes S. phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. PLoS One. 2013;8(4):e61217.

The detailed script of microbial marker identification and POD construction

```
args <- commandArgs(T)

cat("make sure:\n	y and x must have the same sample id\n	train_y must have only two levels and will change to 0 and 1\n	test_y which not in train_y will change to 2\n	set marker_num 0 to compute automaticity\nnote:if x and y have different levels could lead to errors\n")

if (length(args) != 9) {
    stop("Rscript *.R [train_x] [train_y] [test_x] [test_y] [cv_fold] [cv_step] [cv_time] [marker_num] [prefix]\n")
}

train.x <- args[1]
train.y <- args[2]
test.x <- args[3]
test.y <- args[4]
cv.fold <- as.numeric(args[5])
cv.step <- as.numeric(args[6])
cv.time <- as.numeric(args[7])
marker.num <- as.numeric(args[8])
prefix <- args[9]

# package
library(randomForest)

args <- commandArgs(F)
SD <- dirname(sub("--file=", "", args[grep("--file=", args)]))
# function
```

```
source(paste0(SD, "/rfcv1.R"))
source(paste0(SD, "/ROC.R"))

# data
train.x <- t(read.table(train.x))
train.y <- as.factor(read.table(train.y)[, 1])
train.l <- levels(train.y)
levels(train.y) <- 0:1

test.x <- t(read.table(test.x))
test.y <- read.table(test.y)
#test.y <- as.factor(read.table(test.y)[, 1])
#test.l <- levels(test.y)
#levels(test.y) <- pmatch(test.l, train.l) - 1
#test.y <- factor(test.y, 0:2)

# crossvalidation
#pdf.dir <- paste0(prefix, "_randomForest.pdf")
#pdf(pdf.dir, width = 21, height = 7)
#par(mfrow = c(1, 3))

set.seed(0)
train.cv <- replicate(cv.time, rfcv1(train.x, train.y, cv.fold = cv.fold, step = cv.step),
simplify = F)
error.cv <- sapply(train.cv, "[[", "error.cv")
error.cv.rm <- rowMeans(error.cv)
# id <- error.cv.rm < min(error.cv.rm) + diff(range(error.cv.rm))/20
id <- error.cv.rm < min(error.cv.rm) + sd(error.cv.rm)
error.cv[id, ]
if (marker.num == 0) {
    marker.num <- min(as.numeric(names(error.cv.rm)[id]))
}
pdf.dir1=paste0(prefix, "_vars.pdf")
pdf.dir2=paste0(prefix, "_boxplot.pdf")
pdf.dir3=paste0(prefix, "_roc.pdf")
pdf(pdf.dir1)
matplot(train.cv[[1]]$n.var, error.cv, type = "l", log = "x", col = rep(1, cv.time), main
= paste("select", marker.num, "Vars"), xlab = "Number of vars",
    ylab = "CV Error", lty = 1)
lines(train.cv[[1]]$n.var, error.cv.rm, lwd = 2)
abline(v = marker.num, col = "pink", lwd = 2)
dev.off()
# pick marker by corossvalidation
marker.t <- table(unlist(lapply(train.cv, function(x) {
    lapply(x$res, "[", 1:marker.num)
})))
marker.t <- sort(marker.t, d = T)
names(marker.t) <- colnames(train.x)[as.numeric(names(marker.t))]
marker.dir <- paste0(prefix, "_marker.txt")
write.table(marker.t, marker.dir, col.names = F, sep = "\t", quote = F)
marker.p <- names(marker.t)[1:marker.num]

# train model
set.seed(0)
train.rf <- randomForest(train.x[, marker.p], train.y, importance = T)
train.p <- predict(train.rf, type = "prob")
pdf(pdf.dir2)
boxplot(train.p[, 2] ~ train.y, col = 2:3, main = "Probability", names = train.l)
dev.off()
pr.dir <- paste0(prefix, "_train_probability.txt")
```

```
write.table(train.p[, 2], pr.dir, sep = "\t", quote = F, col.names = F)

# train ROC
pdf(pdf.dir3)
plot_roc(train.y, train.p[, 2])
dev.off()
# test predict

test.p <- predict(train.rf, test.x, type = "prob")
pr.dir <- paste0(prefix, "_test_probability.txt")

test.result=cbind(test.y,test.p[,2])
write.table(test.result, pr.dir, sep = "\t", quote = F, col.names = F)

# predict plot
#p.col <- ifelse(is.na(test.y), 4, as.numeric(test.y) + 1)
#plot(rank(test.p[, 2]), test.p[, 2], col = p.col, pch = 16, xlab = "", ylab = "Probability",
main = "Testset")
#txt <- train.l
#if (length(test.l) > 2) {
#    txt <- c(txt, "the rest")
#}
#legend("bottomright", txt, col = 2:4, pch = 16)
#abline(h = 0.5)

# test ROC
#plot_roc(test.y, test.p[, 2])
#dev.off()
```

```
# function
rfcv1 <- function(trainx, trainy, cv.fold = 5, scale = "log", step = 0.5, mtry = function(p) max(1, floor(sqrt(p))), recursive = FALSE,
    ipt = NULL, ...) {
    classRF <- is.factor(trainy)
    n <- nrow(trainx)
    p <- ncol(trainx)
    if (scale == "log") {
        k <- floor(log(p, base = 1/step))
        n.var <- round(p * step^(0:(k - 1)))
        same <- diff(n.var) == 0
        if (any(same))
            n.var <- n.var[-which(same)]
        if (!1 %in% n.var)
            n.var <- c(n.var, 1)
    } else {
        n.var <- seq(from = p, to = 1, by = step)
    }
    k <- length(n.var)
    cv.pred <- vector(k, mode = "list")
    for (i in 1:k) cv.pred[[i]] <- trainy
    if (classRF) {
        f <- trainy
        if (is.null(ipt))
            ipt <- nlevels(trainy) + 1
    } else {
        f <- factor(rep(1:5, length = length(trainy))[order(order(trainy))])
        if (is.null(ipt))
            ipt <- 1
    }
    nlvl <- table(f)
    idx <- numeric(n)
    for (i in 1:length(nlvl)) {
        idx[which(f == levels(f)[i])] <- sample(rep(1:cv.fold, length = nlvl[i]))
    }
    res = list()
    for (i in 1:cv.fold) {
        all.rf <- randomForest(trainx[idx != i, , drop = FALSE], trainy[idx != i], trainx[idx == i, , drop = FALSE], trainy[idx ==
            i], mtry = mtry(p), importance = TRUE, ...)
        cv.pred[[1]][idx == i] <- all.rf$test$predicted
        impvar <- (1:p)[order(all.rf$importance[, ipt], decreasing = TRUE)]
        res[[i]] <- impvar
        for (j in 2:k) {
            imp.idx <- impvar[1:n.var[j]]
            sub.rf <- randomForest(trainx[idx != i, imp.idx, drop = FALSE], trainy[idx != i], trainx[idx == i, imp.idx, drop = FALSE],
                trainy[idx == i], mtry = mtry(n.var[j]), importance = recursive, ...)
            cv.pred[[j]][idx == i] <- sub.rf$test$predicted
            if (recursive) {
                impvar <- (1:length(imp.idx))[order(sub.rf$importance[, ipt], decreasing = TRUE)]
            }
            NULL
        }
        NULL
    }
    if (classRF) {
        error.cv <- sapply(cv.pred, function(x) mean(trainy != x))
    } else {
        error.cv <- sapply(cv.pred, function(x) mean((trainy - x)^2))
    }
    names(error.cv) <- names(cv.pred) <- n.var
    list(n.var = n.var, error.cv = error.cv, predicted = cv.pred, res = res)
}
```